

Eusko Jaurlaritzaren Informatika Elkartea
Sociedad Informática del Gobierno Vasco

SoapUI

Manual de usuario

Fecha:

Referencia:

EJIE S.A.

Mediterráneo, 3

Tel. 945 01 73 00*

Fax. 945 01 73 01

01010 Vitoria-Gasteiz

Posta-kutxatila / Apartado: 809

01080 Vitoria-Gasteiz

www.ejie.es

Control de documentación

Título de documento: SoapUI. Manual de Usuario

Histórico de versiones

Código:

Versión: 2.0

Fecha: 26/3/2010

Resumen de cambios: Cambio Versión SoapUI

Versión: 2.1

Fecha: 26/11/2014

Resumen de cambios: Revisado para SoapUI 5.0

Cambios producidos desde la última versión

Pantallas y opciones de puntos 1, 2 y 3.
Añadido puntos 7

Control de difusión

Responsable: Ander Martínez

Aprobado por: Ander Martínez

Firma: Fecha:

Distribución:

Referencias de archivo

Autor: Consultoría de áreas de conocimiento

Nombre archivo: SoapUI. Manual de usuario v2.1.doc

Localización:


EJIE
Eusko Jaurlaritzaren Informatika Elkartea
Sociedad Informática del Gobierno Vasco

Contenido



1 Introducción

En este manual se describen los distintos aspectos que debe conocer el usuario sobre la herramienta SoapUI.

2 Conceptos básicos

SOAPUI parece que proviene de SOAP y UI. SOAP parecen ser las siglas de siglas de "Simple Object Access Protocol" y UI pueden ser las siglas de "User Interface".

Así, podemos decir que **SoapUI es una herramienta con interface de usuario que permite trabajar con productos software que presenten como interface el protocolo SOAP**.

Para obtener información adicional sobre el producto acceder a su página web:

http://www.soapui.org/

3 SoapUI

3.1 Prerrequisitos

SoapUI es una herramienta sencilla para nuevos usuarios, pero es necesario tener una serie de conocimientos previos para poder sacarle el máximo partido a las funcionalidades de las que SoapUI provee. Es recomendable, aunque no estrictamente necesario, que al usuario tenga conocimientos básico de las siguientes tecnologías:

- **WSDL:** conceptos básicos, tales como servicios, puertos, *"bindings"*, tipos de puerto, relacionados con los esquemas xml.
- **SOAP:** conceptos básicos relacionados con WSDL (*"bindings"*, etc), los diferentes tipos de codificación (*soap-encoded / literal*) y tipos de mensaje (*document / rpc*).
- **XML:** Tanto conocimientos del propio XML como de tecnologías relacionadas, tales como *xpath, XML Schema, namespaces, etc.*

3.2 Interfaz de usuario

SoapUI es una aplicación de escritorio típica, que tiene una interfaz de usuario con estructura similar a la disponible en los IDEs actuales, como pueden ser Eclipse, IDEA o NetBeans. La mayoría de acciones tienen accesos directos o *tooltips*.

La ventana soapUI esta dividida en las siguientes vistas:

1. Parte izquierda: Navegador de proyectos.



2. Parte derecha: Región donde se abren los editores y visualizadores.

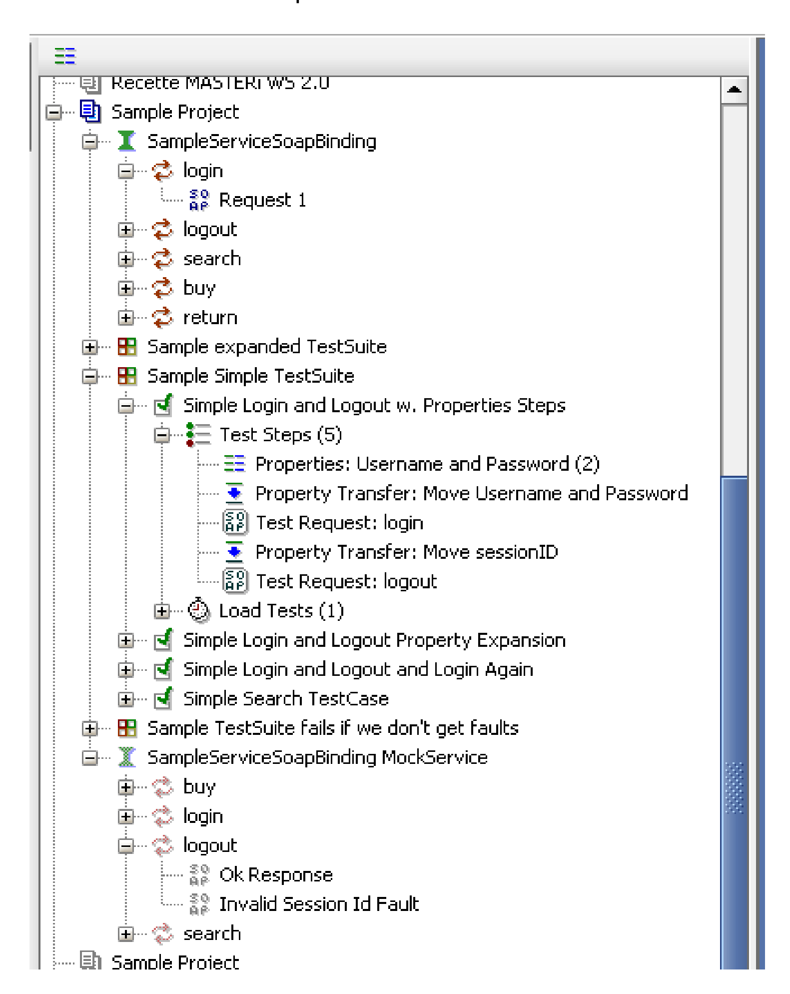
3. Parte inferior izquierda: El panel de propiedades muestra información del objeto seleccionado en el navegador.

4. Parte inferior derecha: Muestra diferentes mensajes de log del soapUI.

3.2.1. El navegador

Los siguiente objetos son mostrados en el árbol de dependencias del navegador:

- **Projects node**: El workspace de soapUI.
  - **Project node(s)**: uno por proyecto en el workspace.
    - **Interface node(s)**: uno para cada interfaz en el proyecto.
      - **Operation node(s)**: uno para cada operación en la interfaz.
        - **Request node(s)**: para cada respuesta creada por una operación.
    - **Nodo(s) TestSuite**: para cada TestSuit en el proyecto.
      - **Nodo(s) TestCase**: para cada TestCase en cada TestSuit.
        - **Nodo TestSteps**: contiene los TestCases
          - **Nodo(s) TestStep**: por cada paso TestCase, junto con un icono coloreado indicando el estado de este paso
        - **Nodo LoadTest**: contiene los TestCases'// antes tb contenia lo mismo pero ahora es con prima, no se si cambiara o no, miralo.
          - **Nodo LoadTest**: para cualquier test de busqueda contenido en el TestCase.
    - **Nodo(s) MockService**: para cada MockService en el proyecto
      - **Nodo(s) MockOperation**: por cada MockOperation en el MockService.
        - **Nodo(s) MockResponse**: por cada MockResponse contenido en un MockOperation.

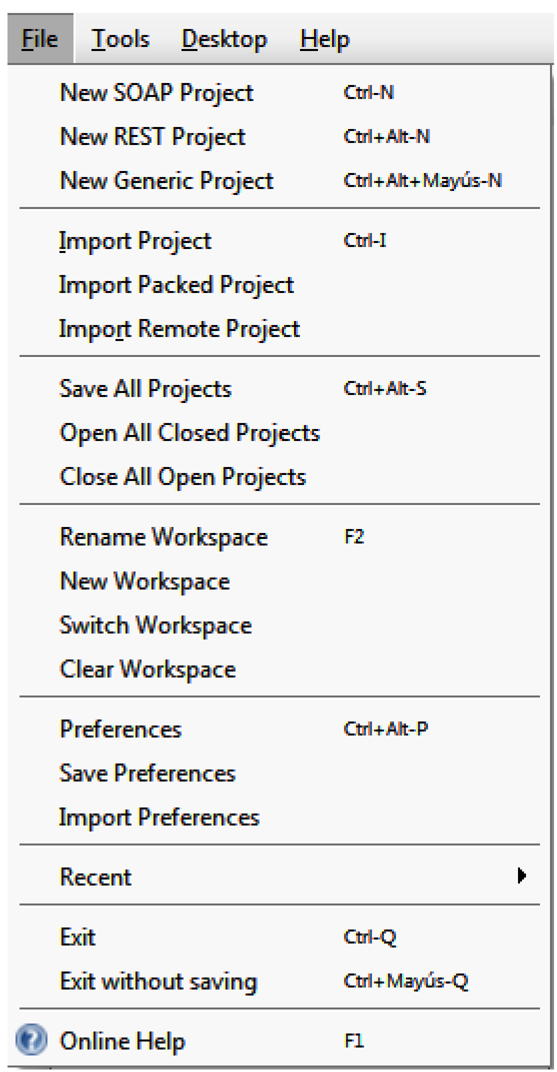


El árbol de dependencias puede ser recorrido usando acciones estándar del teclado. Un objeto asociado en el panel de control puede ser abierto con un doble click o seleccionándolo y presionando enter.

3.2.2. Menú principal

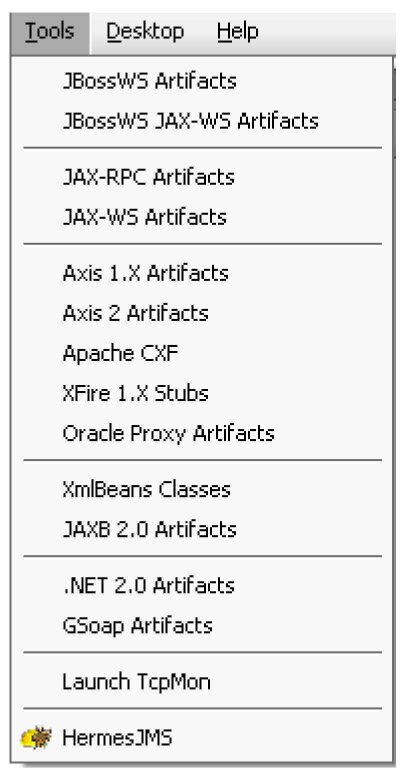
La mayoría de acciones en el soapUI están implementadas a través de botones en la barra de herramientas o mediante click con el botón derecho. Las siguientes opciones están disponibles en el menú principal:

- ***File menu*** (Menú archivo)
  - **New SOPA Project** (Nuevo proyecto SOAP): Inicia el asistente de nuevo proyecto SOAP.
  - **New REST Project** (Nuevo proyecto REST): Inicia el asistente de nuevo proyecto REST.
  - **New Generic Project** (Nuevo proyecto Genérico): Inicia el asistente de nuevo proyecto Genérico.
  - **Import Project** (Importar proyecto): Permite seleccionar el fichero de configuración de un proyecto de SoapUI existente. El proyecto será añadido al workspace existente.
  - **Import Packed Project** (Importar proyecto empaquetado): Permite seleccionar el fichero de configuración de un proyecto de SoapUI existente en formato zip. El proyecto será añadido al workspace existente. Es útil si exportamos un proyecto, que lo hace en este mismo formato.
  - **Import Remote Project** (Importar proyecto remoto): Permite indicar la URL de un proyecto remoto. El proyecto será añadido al workspace existente.
  - **Save All Projects** (Guardar todos los proyectos): Guarda los cambios de todos los proyectos abiertos en el workspace.
  - **Open All Closed Projects** (Abrir todos los proyectos cerrados) – Abre todos los proyectos cerrados en el workspace.
  - **Close All Open Projects** (Cerrar todos los proyectos abiertos): Cierra todos los proyectos abiertos en el workspace. Solicita confirmación.
  - **Rename Workspace** (Renombrar): Renombra el workspace. El nombre se muestra en el nodo raíz del navegador.
  - **New Workspace** (Nuevo Workspace): Permite definir un nuevo workspace.
  - **Switch Workspace** (Cambiar workspace): Permite elegir el workspace que queremos abrir.
  - **Clear Workspace** (Limpia workspace): Permite vaciar el workspace (no borra los proyectos).
  - **Preferences** (Preferencias): Establece las preferencias globales de soapUI.
  - **Save Preferences** (Guardar preferencias): Guarda la configuración global actual.
  - **Import Preferences** (Importar preferencias): Importa configuraciones globales desde otra localización (por ejemplo una instalación anterior de SoapUI). Después de la búsqueda, es necesario reiniciar SoapUI para activar todas las configuraciones.
  - **Recent** (Reciente): contiene submenús con los editores, proyectos y workspaces más recientemente accedidos.
  - **Exit** (Salir): para salir del soapUI.
  - **Exit without saving** (Salir sin guardar): para salir del SoapUI sin guardar.
  - **Online Help** (Ayuda online): lanza la página oficial de documentación en un navegador externo.



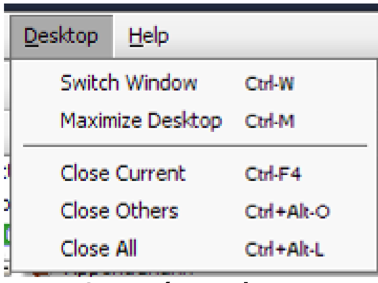
1 Menú "File"

- ***Tools Menu*** (Menú herramientas)
  Contiene acciones para invocar a herramientas externas. Esta integración se descrita en el punto "***¡Error! No se encuentra el origen de la referencia.***".



2 Menú "Tools"

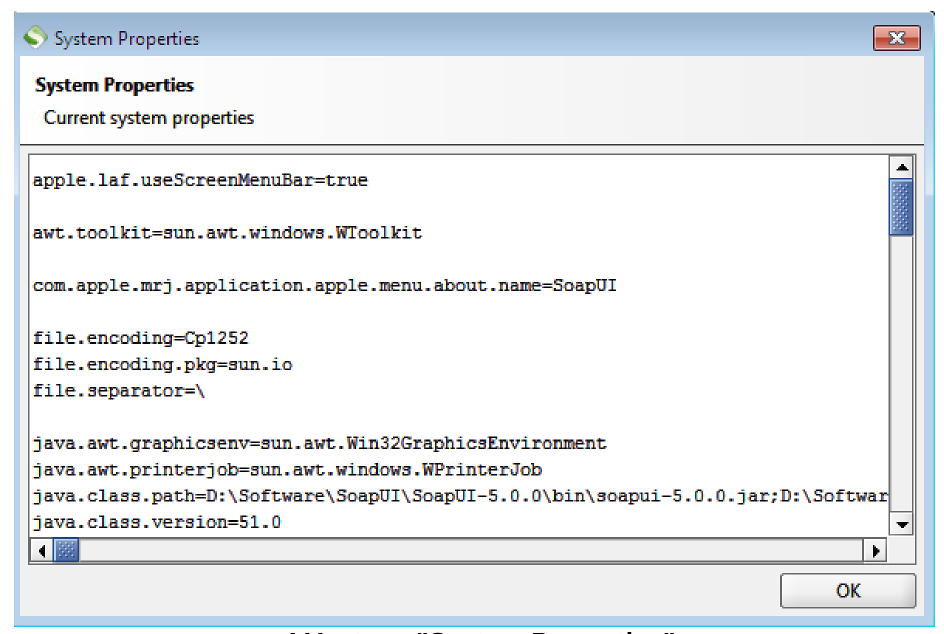
- ***Desktop Menu*** (Menú escritorio)
  Muestra acciones relacionadas con el menu actual.
  - **Switch Windows (**Intercambiar ventanas): abre una ventana para cambiar a otro editor abierto.
  - **Maximize Desktop** (Maximizar escrtorio): oculta/muestra el navegador y las pestañas de log.
  - **Close Current** (Cerrar actual): cierra el panel activo del escritorio.
  - **Close All** (Cerrar todos): cierra todas las vistas abiertas del escritorio.
  - **Closes Others** (Cerrar otros): cierra todas las vistas abiertas del escritorio menos la que en ese momento se encuentre activa.



3 Menú "Desktop"

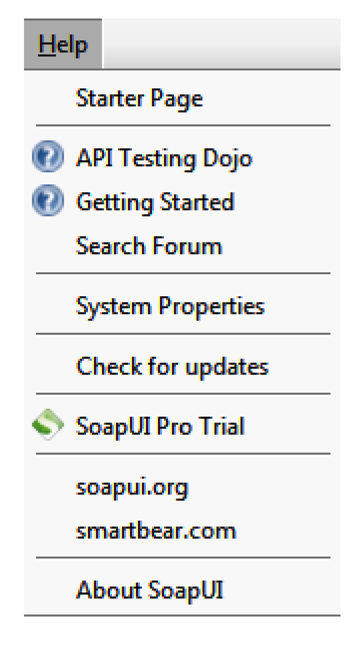
- ***Help Menu*** (Menú ayuda)
  - **Starter Page** (Página Inicial): permite cargar la página inicial.

- **API Testing Dojo** (API de test Dojo): Ayuda en línea del api de test Dojo.
- **Getting started** (Inicio rápido): Abre los documentos de inicio rápido del soapUI
- **Search forum** (Buscar en foros): Permite buscar en foro (requiere acceso Internet)
- **System Properties** (Propiedades del sistema): abre una lista de las propiedades del sistema definidas.



4 Ventana "System Properties"

- **SoapUI Pro Trial:** Acceso a http://www.soapui.org/Downloads/download-soapui-pro-trial.html
- **soapUI.org:** Abre la pagina de inicio del programa soapUI http://www.soapui.org
- **smartbear.com:** Abre la pagina de inicio de la compañía smartbear smartbear.com
- **About SoapUI** (Sobre SoapUI): Muestra información de la versión.

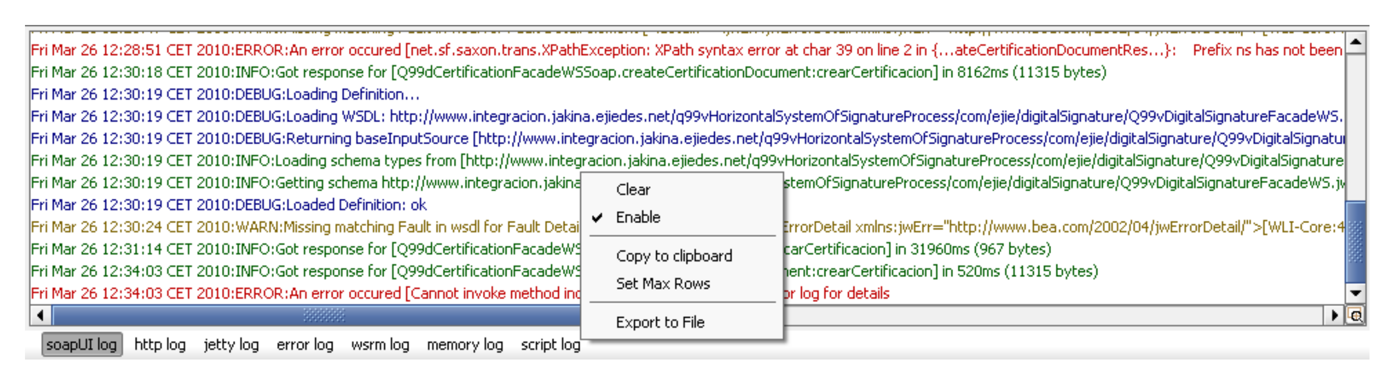




5 Menú "Help"

3.2.3. Pestañas de log

Pulsando el botón derecho en el workspace del soapUI, se pueden observar un cierto número de ventanas de log, cada una mostrando por pantalla la salida correspondiente.



6 Pestaña de log

Al pulsar botón derecho en una pestaña de log, se muestra un menú contextual con opciones para limpiar el log, habilitado o deshabilitarlo, copiar las líneas seleccionadas al portapapeles, etc. También es posible exportar las entradas del log a un fichero. Otra opción es limitar el máximo número de líneas disponibles (por defecto 1000), y que cuando se exceda ese limite, las líneas mas viejas serán borradas del log.

Las diferentes pestañas de log disponibles con las siguientes:

- **SoapUI log**: Notificaciones generales y mensajes.
- **Http log**: Muestra los datos enviados y recibidos por http. Deshabilitado durante las pruebas de stress.
- **Jetty log**: Relacionado con las notificaciones de estado del *mock-service*.
- **Script log**: Los scripts lanzan estos mensajes usando el objeto log disponible (está deshabilitado durante las pruebas de stress pero puede ser habilitado desde /File/Preferencess/UI settings).
- **Error log**: Es un log con información sobre los errores ocurridos durante la ejecución. No tienen porque ser solo errores de soapUI, sino que pueden ser producidos por algún servicio o servidor que no esté disponible.
- **Memory Log:** Muestra información del uso de la memoria**.**

SoapUI utiliza log4j para crear los logs, es posible adaptar la configuración de log4j, renombrando el archivo log4j.xml, llamándolo “soapui-log4j.xml” y posteriormente moviéndolo al directorio *bin* de soapUI.

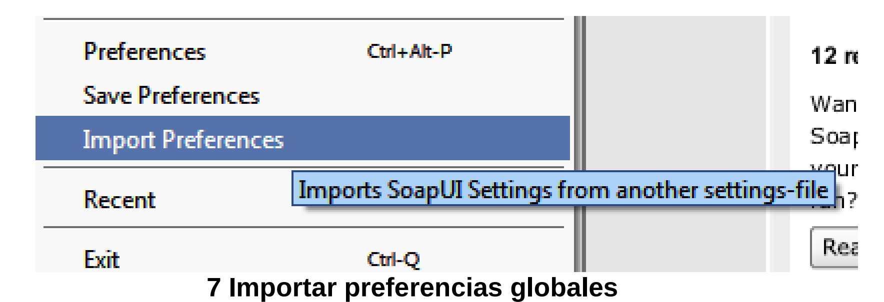
3.3 Preferencias

Estas son las pestañas que se muestran al seleccionar la opción *“Preferences”* en el menú de “*File”*.

Pestaña	Descripción
HTTP Settings	Editar opciones relacionadas con HTTP

Proxy Settings	Editar la dirección http del proxy y la autenticación a utilizar
SSL Settings	Editar opciones relacionadas con SSL
WSDL Settings	Editar opciones relacionadas con WSDL
UI Settings	Editar opciones relacionadas con UI
Editor Settings	Editar opciones relacionadas con el editor
Tools	Editar rutas de las herramientas integradas.
WSI Settings	Editar opciones de validación relacionadas con "*WS-I Basic Profile*"
Global Properties	Administrar propiedades globales.
Global Security Settings	Configuración de una password global, para ocultar la password del proxy en la configuración
WS-A Settings	Configuración de Ws-Addressing
Global Sensitive Information Tokens	Cadenas con información sensible
Version Update Settings	Configuración para actualizar la versión

El punto de menú "File" tiene una acción denominada *"Import preferences"* que permite importar las preferencias globales desde un fichero xml.



7 Importar preferencias globales

3.3.1. HTTP Settings

A continuación se describen los campos configurables de HTTP:

Configuración	Descripción
HTTP Version	Versión de protocolo http.
User-Agent Header	Cambia la cabecera del agente de usuario http. Si no hay ninguna especificada será usada la cabecera HttpClient por defecto.
Request compression	Cambia la cabecera de la petición para indicar que soporta compresión.
Response compression	Para indicar si aceptamos compresión en la respuesta.
Disable Response Decompression	Para deshabilitar la decompresión de respuestas comprimidas.
Close connections after request	Deshabilita el HTTP Keep-Alives pidiendo que se cierre la conexión http después de cada petición. Esto puede tener un efecto negativo en el rendimiento de la aplicación, pero obtendremos unos resultados más realistas.
Chunking Threshold	Valor para aceptar respuestas en “trozos”.
Autenticacion Preemtively	Envía cabeceras de autenticación con cada petición sin haber recibido un desafío de autentificación. Esto es un peligro de seguridad, pero mejora el rendimiento ya que sólo requiere una petición de autenticación en lugar de dos.
Expect-Continue	
Pre-encoded Endopoints	Le indica a SoapUI que no debe realizar un “URL encode” para los endpoints de los servicios web.
Normalize Forward Slashes	
Bind address	La dirección local a utilizar cuando se envían peticiones, puede ser sobrescrita a nivel de petición (con la correspondiente propiedad) y también a nivel de sistema configurando la variable *soapui.bind.address*

Include request in time taken	Incluye el tiempo que se tardo en escribir la petición en el cómputo de tiempo.
Include response in time taken	Incluye el tiempo que se tardo en leer el cuerpo de la respuesta en el cómputo de tiempo.
Socket Timeout	Tiempo máximo para la respuesta http (en milisegundos).
Max Response Size	Máximo número de bytes a leer de una respuesta (0= ilimitado)
Max Connections Per Host	Especifica el número máximo de conexiones a un *host* específico. Incrementar este valor si estas ejecutando LoadTests con más de 500 *threads* en un *host*.
Max Total Connections	Incrementar este valor si estas ejecutando *LoadTests* con más de 2000 *threads*.
Leave MockEngine	Mantiene el *mockEngine* ejecutándose incluso cuando se detiene el *MockService.* Se obtienen mejores tiempos de arranque para nuevos servicios *mock* y además, se obtienen errores 404 al llamar a un servicio *mock* parado, en lugar de un fallo de conexión.
Enable Mock HTTP Log	
Start RestMockService	

HTTP Version:	1.1
User-Agent Header:	
Request compression:	None
Response compression:	[x] Accept compressed responses from hosts
Disable Response Decompression:	[ ] Disable decompression of compressed responses
Close connections after request:	[ ] Closes the HTTP connection after each SOAP request
Chunking Threshold:	
Authenticate Preemptively:	[ ] Adds authentication information to outgoing request
Expect-Continue:	[ ] Adds Expect-Continue header to outgoing request
Pre-encoded Endpoints:	[ ] Do not URL-escape service endpoints
Normalize Forward Slashes:	[ ] Replaces duplicate forward slashes in HTTP request endpoints with a single slash
Bind Address:	
Include request in time taken:	[x] Includes the time it took to write the request in time-taken
Include response in time taken:	[x] Includes the time it took to read the entire response in time-taken
Socket Timeout (ms):	250000
Max response size:	
Max Connections Per Host:	
Max Total Connections:	2000
Leave MockEngine:	[x] Leave MockEngine running when stopping MockServices
Enable Mock HTTP Log:	[ ] Logs wire content of all mock requests
Start RestMockService:	[x] Start RestMockService after creation

8 Configuración HTTP por defecto

3.3.2. Proxy Settings

A continuación se describen los parámetros de configuración para **proxy**:

Opción	Descripción
Host del proxy	Indicar el proxy http a usar
Puerto del proxy	Indicar el puerto del proxy http a usar
Nombre de usuario del proxy	Nombre de usuario para autenticarse en el proxy
Contraseña de usuario del proxy	Contraseña de usuario para la autenticase en el proxy

Excluidos	Una lista de hosts a excluir separadas por coma. Ejemplo: “127.0.0.1:8080, myserver.com” no usara un proxy desde 127.0.0.1 en el puerto 8080 y myserver en ningún puerto.

Proxy Setting: Automatic
None
Manual
Host: intercon.ejgvdns
Port: 8080
Excludes: letza.jaso,xlnets.servicios.jakina.ejiedes.net,www.otc.ejiedes.net
Username: jriobell
Password: ••••••

9 Configuración de proxy por defecto

3.3.3. SSL Settings

A continuación se describen los parámetros de configuración para **SSL**:

Configuración	Descripción
keyStore	Path al *keystore* con los certificados de cliente, o trusted de servidor para ssl en un lado.
keyStore password	Password del keystore..
Enable Mock SSL	Habilita el soporte para SSL en *MockServices*
Mock Port	Puerto para las conexiones SSL.
Mock KeyStore	Keystore a utilizar para los certificados SSL
Mock Password	Password del keystore.
Mock Key Password	Password por defecto para las claves.
Mock TrustStore	Truststore a utilizar (opcional).
Mock TrustStore Password	Password para el truststore.
Client Authentication	Marcar si se requiere certificado de cliente

KeyStore: [Browse...]
KeyStore Password:
Enable Mock SSL: enable SSL for Mock Services
Mock Port:
Mock KeyStore: [Browse...]
Mock Password:
Mock Key Password:
Mock TrustStore: [Browse...]
Mock TrustStore Password:
Client Authentication: requires client authentication

10 Ejemplo de configuración SSL

3.3.4. WSDL Settings

A continuación se describen los parámetros de configuración para **WSDL**:

Configuración	Descripción
Cache WSDLs	Habilita/deshabilita la caché de WSDLs.
Sample Values	Genera valores de muestra en las peticiones si se dispone del esquema.
Type Comment	Genera comentarios con información del tipo en nuevas peticiones
Include Optional	Incluir siempre elementos opcionales en las peticiones generadas.
Pretty Print	Formatear los mensajes de respuesta en el editor de respuestas.
Attachment Parts	Genera *part-elements* en los mensajes de petición para los *mime-attachments* de los mensajes RPC (requerido por algunos *ws-stacks*).
No Content-Type Validation	No valida el *content-type* de un *mime-attachment* contra el tipo especificado en el *SOAP-Binding*

Schema Directory	Especifica el directorio que contiene los ficheros de esquemas (.xsd) que deben ser añadidos automáticamente cuando se parseen o validen wsdls o peticiones. Cambiar el valor de este directorio requiere reiniciar el programa.
Name with Binding	Le dice a soapUI que denomine las interfaces importadas con el nombre correspondiente a su *soap/http binding*, y no con su *portType*. Esto asegura que los WSDLs que contienen *bindings* tanto para soap 1.1 como para soap 1.2, tendrán nombres únicos durante la importación. Por defecto tiene valor “true”.
Excluded Types	Es una lista de tipos y elementos globales de XML-Schema, de la forma name@namespace, que serán usados cuando se generen peticiones y respuestas de muestra.
Strict Schema Types	Seleccionar esta opción para no permitir redefiniciones de los tipos de esquema en los XSDs incluidos/importados de un WSDL específico.
Compression Limit	Es el tamaño mínimo de un mensaje para ser comprimido y preservar el espacio. La compresión es realizada mediante gzip y el resultado esta codificado en base64 en el archivo del proyecto. Para peticiones grandes, puede ahorrar más de un 90% de espacio en el fichero, pero las peticiones dejan de ser entendibles y comparables.
Pretty Print Project Files	Formatea los ficheros de proyecto cuando son guardados. Hace más sencillo trabajar con un sistema de control de versiones (SCM). Esta opción incrementa el tamaño de los ficheros de proyecto. También se formatearán los ficheros de caché para WSDL/XSD.
Trim WSDL	Elimina espacios en blanco antes y después del WSDL.

Cache WSDLs:	[x] caches and associated WSDLs locally for offline access and improved performance
Sample Values:	[ ] generate example values in new requests
Type Comment:	[ ] generate comments with type information in new requests
Include Optional:	[x] always include optional schema elements when creating requests
Pretty Print:	[x] pretty print response messages
Attachment Parts:	[ ] generate rpc message parts for attachments
No Content-Type Validation:	[ ] allow incorrect content-types in mime-attachments
Schema Directory:	Browse...
Name with Binding:	[x] uses the WSDL binding name (instead of portType) for imported Interfaces
Excluded types:	schema@http://www.w3.org/2001/XMLSchema — Add.. Edit.. Remove..
Strict schema types:	[ ] fails schema imports if types/particles are redefined
Compression Limit:	
Pretty Print Project Files:	[ ] pretty prints project files
Trim WSDL:	[ ] Trims leading and trailing whitespaces from WSDL file (might not work on non 8 bit encoding)

11 Configuración WSDL por defecto

3.3.5. UI Settings

A continuación se describen los parámetros de configuración relativos a la interfaz de usuario:

Configuración	Descripción
Cerrar proyectos	Cierra todos los proyectos en el arranque para mejorar el tiempo de arranque y minimiza el consumo de memoria.
Ordenar proyectos	Ordena los proyectos en orden alfabético en el navegador.
Ordenar Servicios	Ordena los servicios en orden alfabético en el navegador.
Ordenar peticiones	Ordena las peticiones en orden alfabético en el navegador.
Mostrar descripciones	Muestra la descripción si está definida
Grabar proyectos al salir	Graba el proyectos al salir de soapui
Crar un backup	Hace un backup del proyecto antes de guardarlo
Directorio de backups	

Intervalo de autoguardado	Indica el intervalo máximo (en minutos) para un proyecto sin que se ejecute el autoguardado. Si esta a 0, soapUI salvara automáticamente todos los proyectos que no tengan tests ejecutándose en el intervalo especificado.
Tipo de escritorio	Indica el diseño del escritorio a utilizar. Los cambios de configuración serán aplicados cuando cerremos la ventana de preferencias.
Seleccionar el más reciente usado panel de escritorio al salir	
Native L&F	Deshabilita el uso del *look & feel* por defecto, y usa el especificado por la JRE.
No deshabilitar el log de Groovy	Mantiene activo el log de groovy durante los LoadTests, el cual puede ser usado para debugear, etc.
Mostrar pestaña de log	Expande las pestañas de log al iniciar el soapUI.
Mostrar Pagina de Inicio	
Deshabilitar tooltips	
Normalizar cambios de linea	
Intervalo en segundos del recolector de basura	
Tamaño en RAW de la respuesta a mostrar	
Tamaño en RAW de la petición a mostrar	
Cortar el contenido en trozos	

Close Projects:	☐ Close all projects on startup
Order Projects:	☐ Order Projects alphabetically in tree
Order Services:	☐ Order Services alphabetically in tree
Order Requests:	☐ Order Requests alphabetically in tree
Show Descriptions:	☑ Show description content when available
Save projects on exit:	☑ Save all projects on exit
Create Backup:	☐ Backup project files before they are saved
Backup Folder:	
AutoSave Interval:	0
Desktop Type:	Default
Select most recently used desktop panel on close:	☐ Show most recently used panel on close (requires restart)
Native LF:	☐ Use native Look & Feel (requires restart)
Do not disable Groovy Log:	☐ Do not disable the groovy log when running load tests
Show Log Tabs:	☐ Shows log tabs when starting SoapUI
Show Startup Page:	☑ Opens startup web page when starting SoapUI
Disable Tooltips:	☐ Disables all tooltips
Normalize line-break:	☐ Normalize line-breaks when saving project
Garbage Collection Interval (s):	60
Size of Raw Response Message to Show:	10000
Size of Raw Request Message to Show:	10000
Wrap content in Raw Message Viewers:	☐ Wraps content in Raw Message Viewers

12 Configuración de UI

3.3.6. Editor Settings

A continuación se describen los parámetros de configuración relativos al editor:

Configuración	Descripción
Fuente del editor	Especifica la fuente a usar por el editor XML. Al pulsar este botón se abrirá un dialogo en el que se puede seleccionar el tipo de letra y el tamaño deseado.
Numero de líneas en el XML	Muestra por defecto los numero de las líneas en todos los editores XML (Alt+L en los editores para cambia esta opción).

Numero de líneas Groovy	Muestra por defecto los números de las líneas en todos los editores Groovy (Alt+L en los editores cambia esta opción).
Deshabilitar auto-redimensionado	Deshabilita el redimensionado automático en los editores de peticiones / respuestas.
Vista de la pestaña de petición	Pone el diseño de la pestaña como diseño por defecto para los editores de peticiones / respuestas.
Validar peticiones	Habilita la validación automática de las peticiones antes de que sean presentadas desde un editor de peticiones. Seleccionar esta opción es lo mismo que usar Alt-V en el editor.
Validar respuestas	

Editor Font: Monospaced.plain 11 [Select Font..]
XML Line Numbers: Show line numbers in XML editors by default
Groovy Line Numbers: Show line numbers in Groovy editors by default
Disable auto-resize: Disables automatic resizing of Request editors
Tabbed request view: Defaults the Request editor to the tabbed layout
Validate Requests: Always validate request messages before they are sent
Abort on invalid: Abort invalid requests
Validate Responses: Always validate response messages

13 Configuración editor por defecto

3.3.7. Integrated Tools

Aquí se indican los *paths* a las herramientas externas que deseamos integrar en SoapUI.

JBossWS wstools:		Browse...
JAX-RPC WSCompile:		Browse...
JAX-WS WSImport:	D:\Software\jwsdp-2.0\jaxws\bin	Browse...
Axis 1.X:		Browse...
Axis 2:		Browse...
.NET 2.0 wsdl.exe:		Browse...
XFire 1.X:		Browse...
CXF 2.X:		Browse...
ANT 1.6+:		Browse...
GSoap:		Browse...
JAXB xjc:		Browse...
XmlBeans 2.X:		Browse...
JDK 1.5 javac:		Browse...
Apache TcpMon:		Browse...
Oracle wsa.jar:		Browse...
WADL2Java:		Browse...
Hermes JMS:		Browse...

14 Configuración de paths de herramientas externas

3.3.8. WSI Settings

Estos ajustes están relacionados con la validación WS-I.

Ajustes	Descripción
Verbose	Establece la salida *verbose* de la herramienta WS-I
Results Type	Establece que resultados mostrará en el informe generado.
Message Entry	Muestra las entradas del mensaje en el informe
Failure Message	Incluye los mensajes de fallo definidos en el informe
Assertion Description	Incluye la descripción de cada *test-assertion* en el informe.

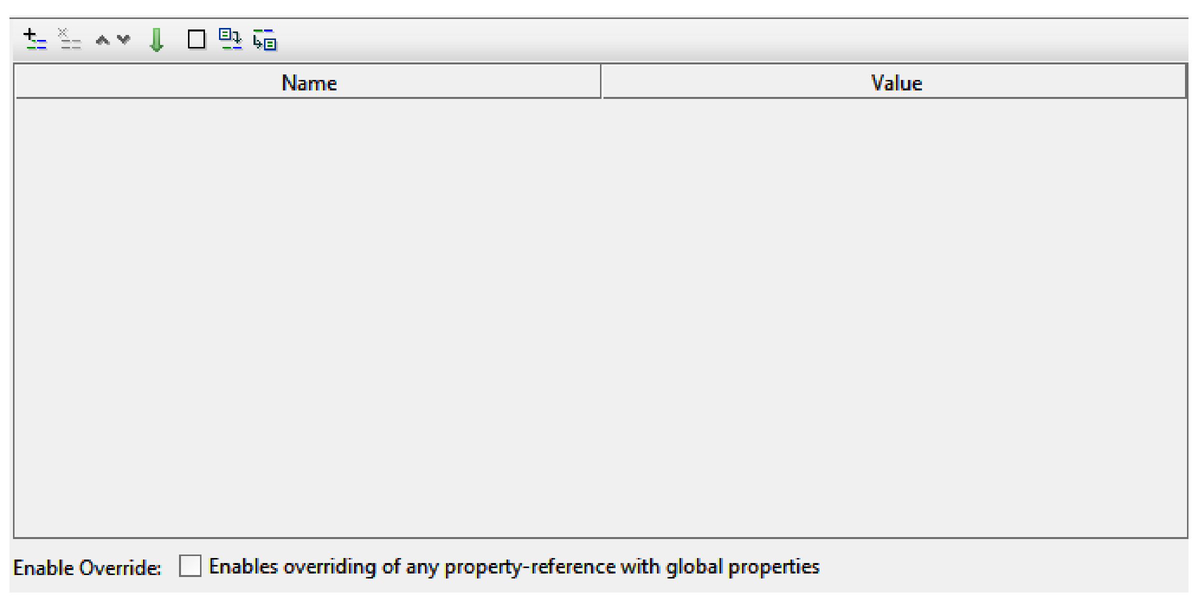
Tool Location	Path a la instalación de las herramientas de testeo WS-I.
Show Log	Muestra la ventana de log cuando se ejecutan herramientas WS-I.
Output Folder	Si esta seleccionado, los informes HTML generados serán exportados automáticamente a esta carpeta.

Verbose: sets verbose output of WSI tools
Results Type: all
Message Entry: if message entries should be included in the report
Failure Message: if failure message defined for each test assertion should be included in the report
Assertion Description: if description of each test assertion should be included in the report
Tool Location: d:\software\soapui\soapUI-3.6.1/wsi-test-tools Browse...
Show Log: show console-log for ws-i analyzer
Output Folder: Browse...

15 Configuración de WS-I por defecto

3.3.9. Global Properties

Administra la propiedades globales, si un archivo de propiedades SoapUI ha sido especificado, los contenidos de estos archivos estarán disponibles en la tabla. La opción de “habilitar sobreescritura” puede hacer que las propiedades globales siempre sobrescriban otras propiedades referenciadas.



16 Configuración de propiedades globales por defecto

3.3.10. Global Security Settings

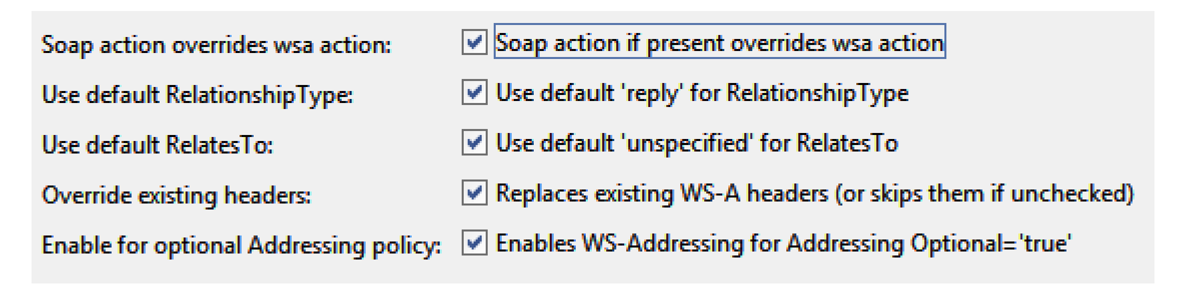
Se puede incluir una clave general para codificar las claves en los ficheros de configuración, que por defecto son legibles.



17 Configuración de propiedades globales de seguridad

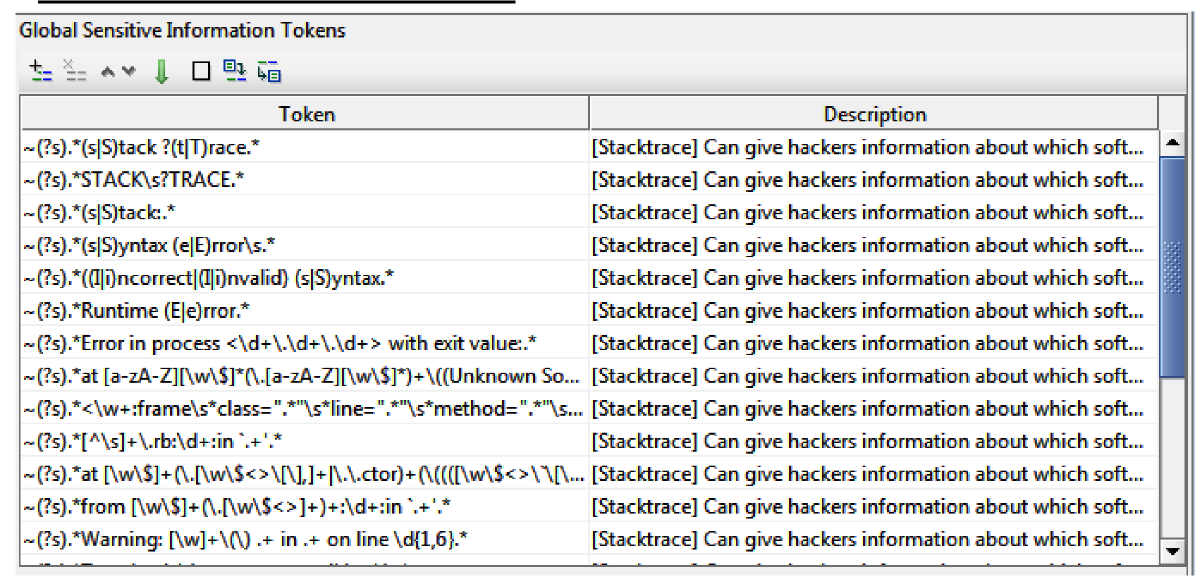
3.3.11. WS-A Settings

WS-A es un estándar de WS que permite incluir directivas de enrutamiento en el propio protocolo SOAP.



18 Configuración de WS-A Settings

3.3.12. Global Sensistive Information Tokens



Global Sensitive Information Tokens

<table>
<tr><th>Token</th><th>Description</th></tr>
<tr><td>~(?s).*(s|S)tack ?(t|T)race.*</td><td>[Stacktrace] Can give hackers information about which soft...</td></tr>
<tr><td>~(?s).*STACK\s?TRACE.*</td><td>[Stacktrace] Can give hackers information about which soft...</td></tr>
<tr><td>~(?s).*(s|S)tack:.*</td><td>[Stacktrace] Can give hackers information about which soft...</td></tr>
<tr><td>~(?s).*(s|S)yntax (e|E)rror\s.*</td><td>[Stacktrace] Can give hackers information about which soft...</td></tr>
<tr><td>~(?s).*((I|i)ncorrect|(I|i)nvalid) (s|S)yntax.*</td><td>[Stacktrace] Can give hackers information about which soft...</td></tr>
<tr><td>~(?s).*Runtime (E|e)rror.*</td><td>[Stacktrace] Can give hackers information about which soft...</td></tr>
<tr><td>~(?s).*Error in process <\d+\.\d+\.\d+> with exit value:.*</td><td>[Stacktrace] Can give hackers information about which soft...</td></tr>
<tr><td>~(?s).*at [a-zA-Z][\w\$]*(\.[a-zA-Z][\w\$]*)+\((Unknown So...</td><td>[Stacktrace] Can give hackers information about which soft...</td></tr>
<tr><td>~(?s).*<\w+:frame\s*class=".*"\s*line=".*"\s*method=".*"\s...</td><td>[Stacktrace] Can give hackers information about which soft...</td></tr>
<tr><td>~(?s).*[^\s]+\.rb:\d+:in `.+'.*</td><td>[Stacktrace] Can give hackers information about which soft...</td></tr>
<tr><td>~(?s).*at [\w\$]+(\.[\w\$<>\[\],]+|\.\.ctor)+(\((([\w\$<>\`\[\...</td><td>[Stacktrace] Can give hackers information about which soft...</td></tr>
<tr><td>~(?s).*from [\w\$]+(\.[\w\$<>]+)+:\d+:in `.+'.*</td><td>[Stacktrace] Can give hackers information about which soft...</td></tr>
<tr><td>~(?s).*Warning: [\w]+\(\) .+ in .+ on line \d{1,6}.*</td><td>[Stacktrace] Can give hackers information about which soft...</td></tr>
</table>

19 Configuración de Global Sensitive Information Tokens

3.3.13. Version Update Settings

Check for new versions: ☐ Automatically check for new versions of SoapUI

20 Configuración de Version Update Settings

4 Pruebas funcionales

SoapUI soporta pruebas funcionales de Servicios Web suministrando un caso de prueba con un número de pasos que pueden ser ejecutados en secuencia. En la actualidad, hay seis tipos de pasos que proporcionan muchas posibilidades de prueba. Los casos de prueba están organizados en un grupo de pruebas y en un mismo proyecto se pueden crear varios grupos de pruebas.

Las pruebas funcionales, en soapUI, se pueden usar para una variedad de propósitos:

- Unidad de prueba: valida que cada operación del Servicio Web funciona como se indica
- Pruebas de compatibilidad: valida que el resultado devuelto por el Servicio Web es compatible con su definición
- Prueba de procesos: valida que una secuencia de invocaciones de Servicios Web ejecuta un proceso de negocio requerido
- Pruebas guiadas por datos: valida que cualquiera de los anteriores funciona como requerimiento de datos de entrada procedentes de fuentes externas (por ejemplo, una base de datos u otro servicio Web)

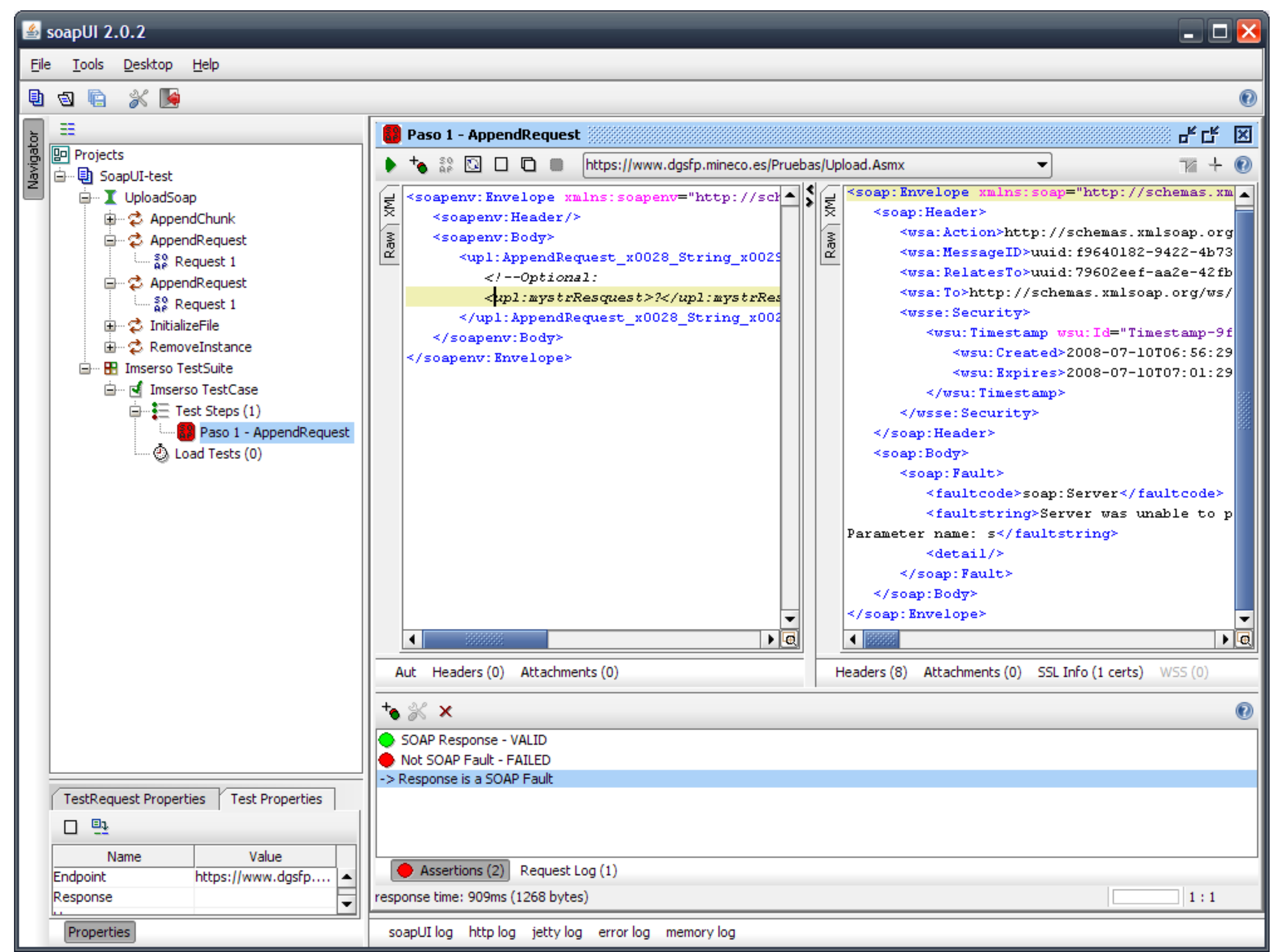
4.1 Introducción

En soapUI, las pruebas funcionales se pueden usar para validar requisitos funcionales, tanto para invocar los Servicios Web propios (=”pruebas unitarias”) como para una secuencia de peticiones (= " pruebas de integración"). Además, es posible añadir lógica a las pruebas mediante *scripts* de *Groovy,* lo que permite, por ejemplo, interactuar con una base de datos o realizar un flujo de pruebas complejo.

Crear un Caso de Prueba desde una petición

Una vez que tienes algunas peticiones configuradas, puedes crear un Caso de Prueba que verifique su comportamiento:

- Selecciona el segundo botón de la barra de herramientas en la ventana de edición de petición. ("*Add this request to a test-case*")
- Si no hay ningún Grupo de Prueba/Caso de Prueba en tú proyecto, soapUI abrirá una ventana para pedirle los nombres de éstos, puedes especificar algo como "Imserso TestSuite" y "Imserso TestCase"
- SoapUI abrirá una ventana para introducir el nombre de su petición de prueba, por ejemplo “Paso 1 - AppendRequest”.
- Los correspondientes TestSuite/TestCase se crearán y la petición se añadirá como TestCase, la cuál es una copia de la petición original (de esta manera, puedes seguir utilizándolo sin modificar la petición de prueba)
- Un editor de peticiones de prueba casi idéntico al estándar, se abre con el nuevo TestCase; difiere por añadir aserciones de funcionalidad.

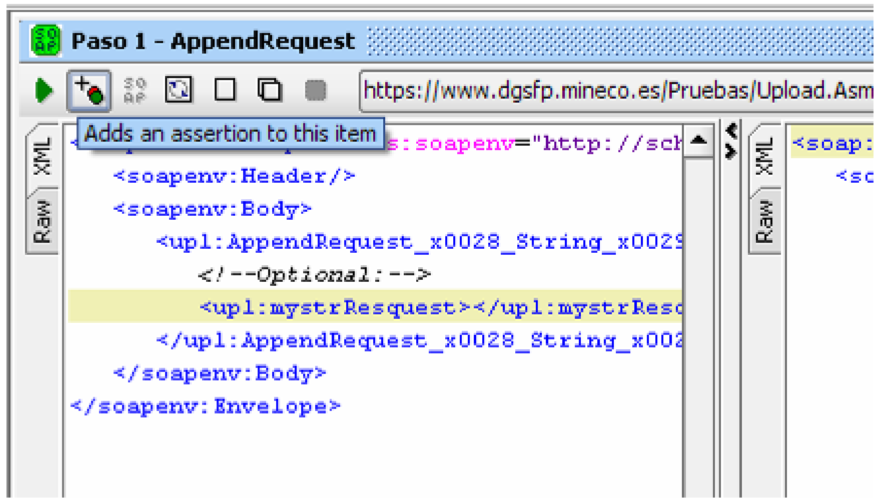


21 Ejemplo de ejecución de un TestCase

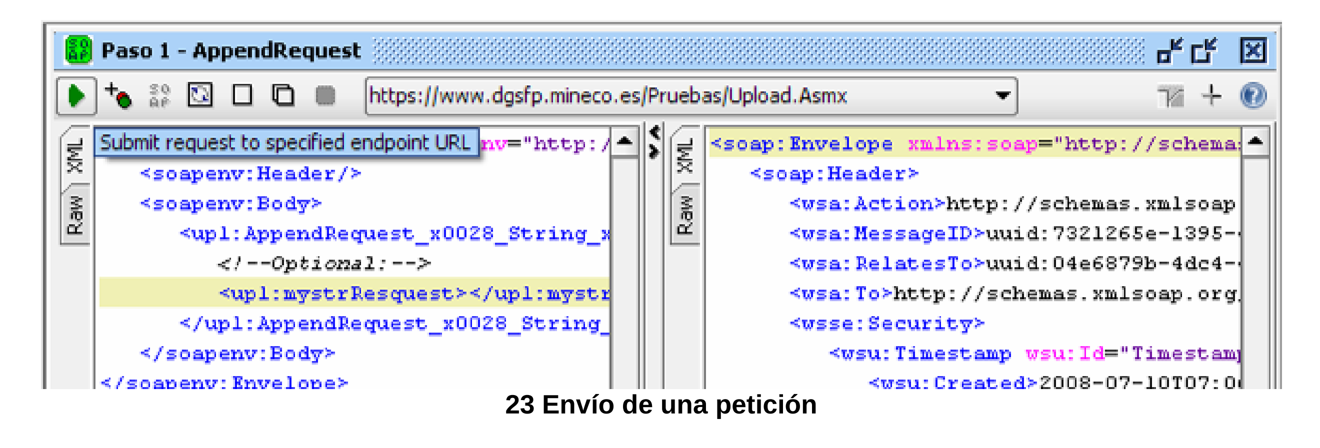
Añadir aserciones

Las aserciones son condiciones que se evalúan tras obtener la respuesta y que su cumplimiento indica que un funcionamiento correcto:

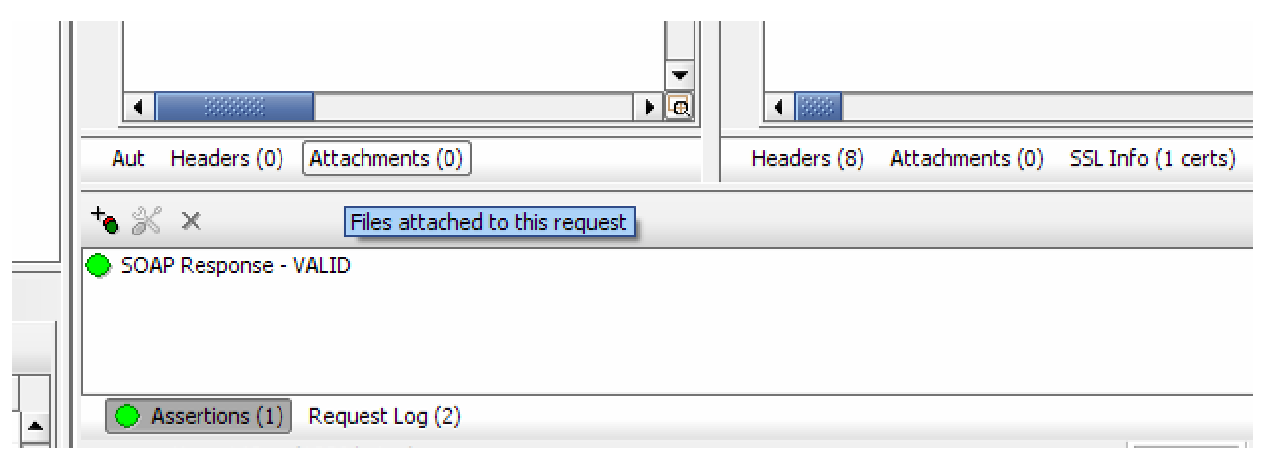
- Selecciona el segundo botón de la barra de herramientas en la ventana de edición de la petición ("Add an assertion to this test request").
- Empieza añadiendo una aserción de conformidad de esquema, que se encargará de comprobar que la petición es compatible con las asociaciones definidas en el esquema WSDL. La aserción se mostrará en la lista de aserciones debajo de los editores de petición/respuesta. (mirar imagen inferior)
- La petición es enviada haciendo click en el botón verde ubicado en la parte superior izquierda. SoapUI lanzará la petición y validará la respuesta. Si todo va bien la petición-prueba debería ser presentada con un fondo verde en el árbol de navegación.



22 Añadir aserción a peetición de un TestCase



23 Envío de una petición



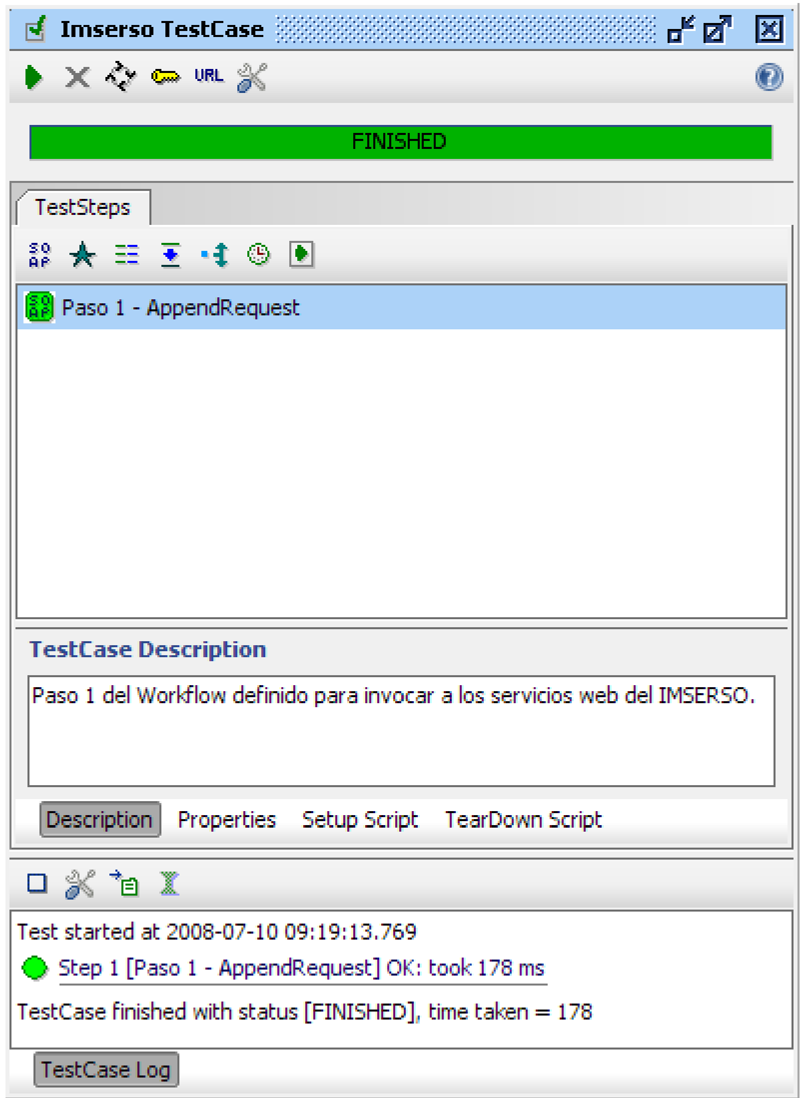
24 Resumen del resultado aserciones de las aserciones

Lanzar TestSuite

Una vez que están todas los *TestCase* con sus aserciones, es posible lanzar todo el caso de prueba, para ello:

- Doble click en el nodo del TestCase en el cuadro izquierdo de navegación, que abrirá el lanzador del caso de prueba.
- Lanza todas las pruebas haciendo click en la flecha verde etiquetada con "*Run this TestCase*", SoapUI enviará todas las peticiones de prueba y las validará consecuentemente. El resultado se mostrará durante la ejecución.

Lanza tus pruebas desde la línea de comandos usando uno de las herramientas de línea de comando disponibles.



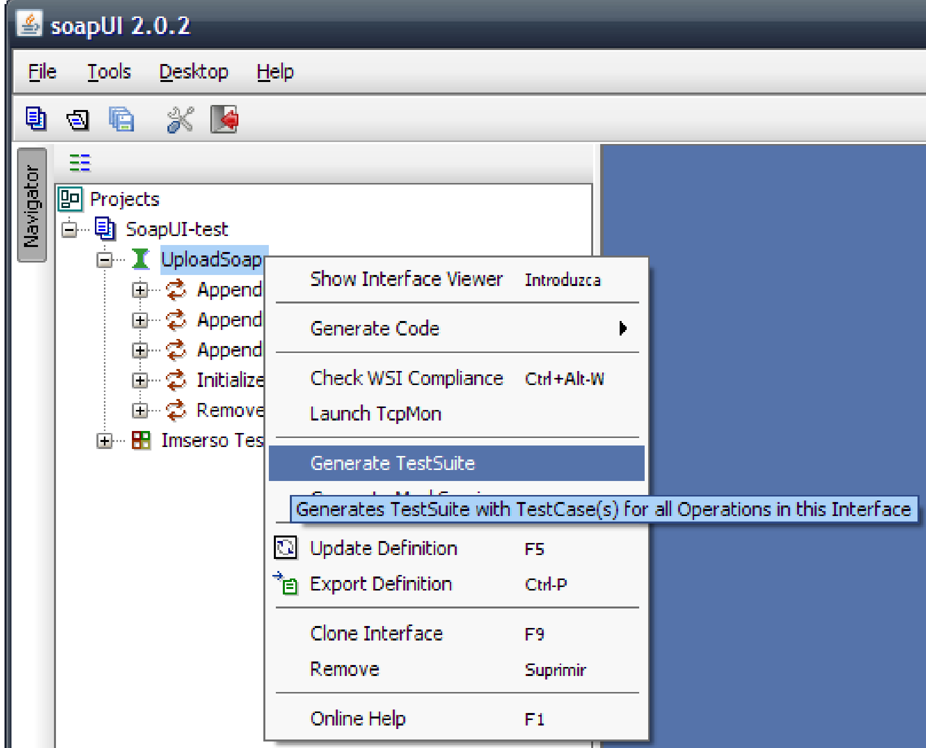
25 Resultados Imserso TestCase

4.2 TestSuites

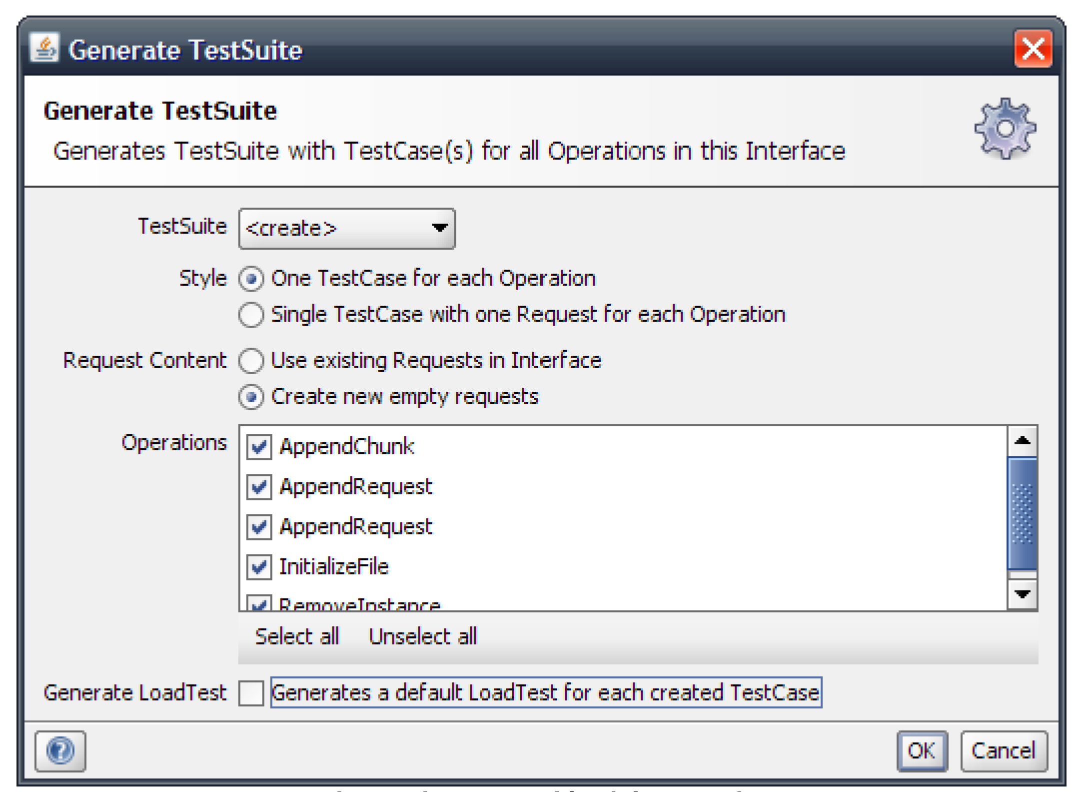
Un TestSuite sirve como contenedor para un número arbitrario de TestCase. Cuando se ejecuta un TestSuite, los TestCase incluidos se pueden ejecutar tanto en secuencia como en paralelo, tal y como se describe a continuación.

Crear TestSuites

Selecciona la opción de "Generate TestSuite" de la interfaz del menú para abrir una ventana y generar un TestSuite completo para la interfaz seleccionada. El cuadro de diálogo contiene las siguientes opciones:



26 Crear un TestSuite nuevo



27 Opciones de generación del TestSuite

- **TestSuite**: selecciona si quieres crear un TestSuite en uno existente o en uno nuevo.
- **Style**: hay 2 tipos diferentes de estilos:
  1. *One TestCase for each Operation*: crea un TestSuite con un Caso de Prueba para cada operación
  2. *Create new Empty Requests*: crea una petición vacía con contenido opcional en las peticiones de prueba creadas
- **Operations**: selecciona las operaciones que quieres que se generen.
- **Generate LoadTest:** crea una Prueba de Carga por defecto para cada Caso de Prueba generado.

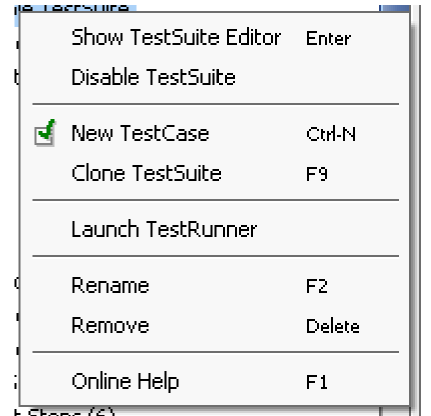
Acciones del TestSuite

Las siguientes acciones están disponibles desde del menú que aparece al realizar clic derecho sobre el nodo del TestSuite.

- **Open TestSuite Editor**: abre el lanzador del TestSuite. Se describe más abajo.
- **Disable/Enable TestSuite**: deshabilita/habilita el TestSuite.
- **New TestCase**: aparece una ventana para definir un nuevo Caso de Prueba en el Grupo de Prueba.
- **Clone TestSuite**: genera una ventana para clonar un TestSuite, incluyendo todos los TestCase y TestSteps.
- **Launch TestRunner**: abre un cuadro de diálogo para ejecutar el TestSuite desde la línea de comandos.
- **Rename**: abre una ventana para renombrar el TestSuite.

Remove: aparece una ventana para borrar el TestSuite del proyecto. Todos los TestSuite que contiene también serán borrados

- **Online Help**: abre la página de ayuda online en el navegador.

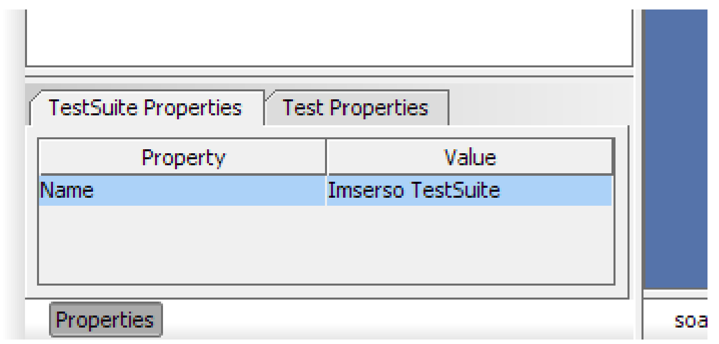


28 Menú contextual nodo TestSuite

Detalles de la etiqueta de TestSuite

La etiqueta de “detalles” que está en la esquina inferior izquierda, muestra los siguientes valores cuando el TestSuite está seleccionado en el árbol de navegación.

- **Name**: el nombre del TestSuite actual



29 Propiedades TestSuite

Lanzador de TestSuite

Haciendo doble click en el navegador del TestSuite, abre el lanzador de TestSuite que contiene una lista de los TestCase y una barra de herramientas. Se muestra una barra de progreso para cada TestCase. Al hacer doble clic en un TestCase abre su editor de TestCase. Si un TestCase está probando su carga, su barra de progreso lo muestra y el TestCase no se lanzará durante la ejecución del TestSuite.

Los botones disponibles en la barra de herramientas de izquierda a derecha son:

- **Run**: ejecuta los TestCase seleccionados.
- **Cancel**: cancela la ejecución en curso.

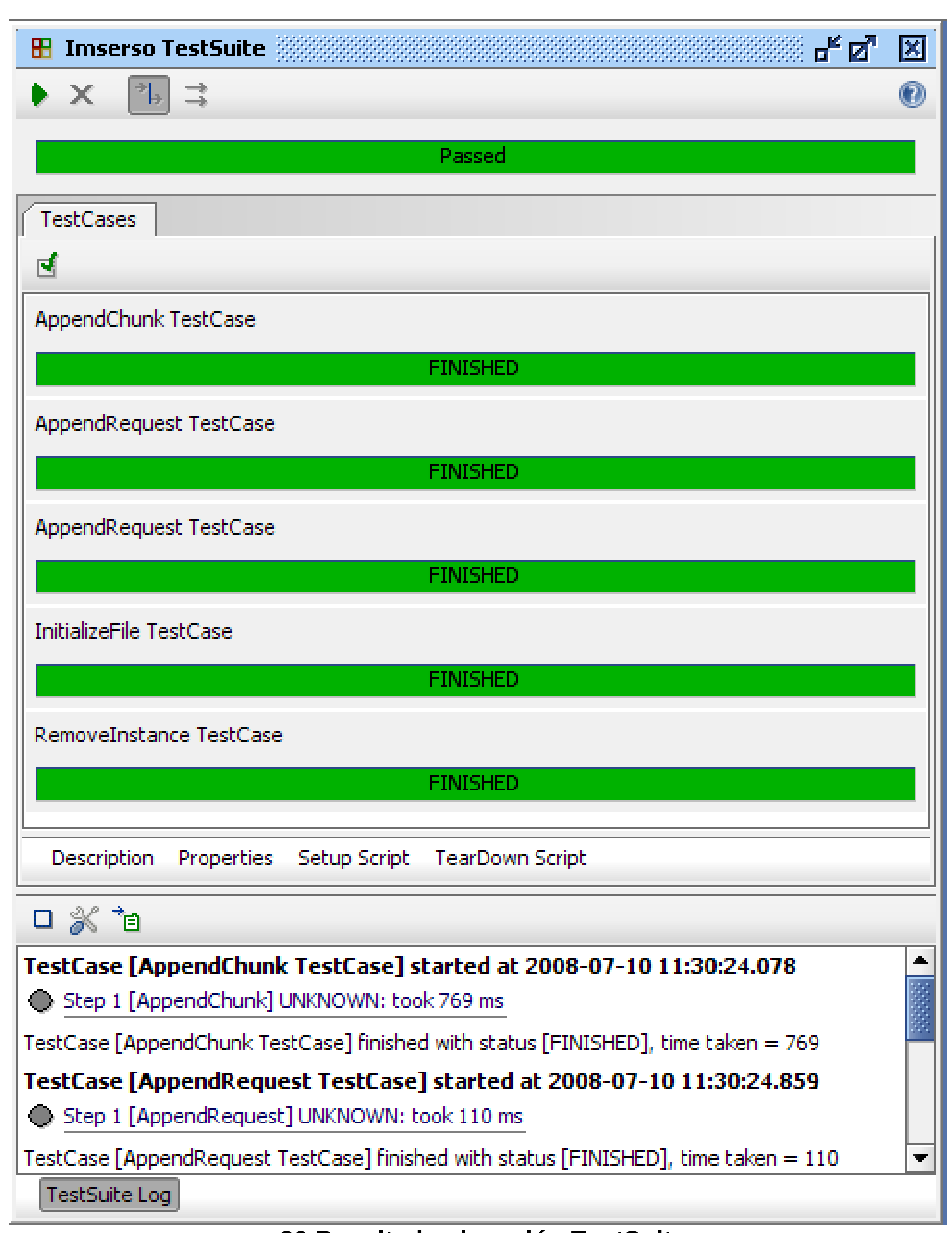
- **New TestCase**: abre una ventana para crear un nuevo TestCase en este TestSuite.
- **Run in Sequence**: determina si el TestCase debe ser ejecutado en secuencia.
- **Run in Parallel**: determina si el TestCase debe ser ejecutado en paralelo.

El estado de ejecutar en secuencia/paralelo es conservado y también se aplica cuando se ejecuta un TestSuite usando una de las herramientas de línea de comando o un plugin experto.

Después de la barra de progreso viene una lista de los TestCase que contiene, seguido por una serie de inspectores para el TestCase (de izquierda a derecha):

- **Description**: una descripción parcial para el TestSuite.
- **Properties:** propiedades del TestSuite.
- **Setup Script**: un script para ejecutar cuando el TestCase se está ejecutando.
- **TearDown Script**: un script para ejecutar cuando el TestCase termine.

La ventana de registro del TestSuite de abajo, muestra todos los pasos/ejecuciones resultados del último TestSuite lanzado.



30 Resultado ejecución TestSuite

4.3 TestCase

SoapUI soporta pruebas funcionales de Servicios Web suministrando un TestCase con un número de pasos que pueden ser ejecutados en secuencia. Además, un número arbitrario de pruebas de carga se pueden asociarse con un TestCase para ejecutarse bajo diferentes escenarios de carga.

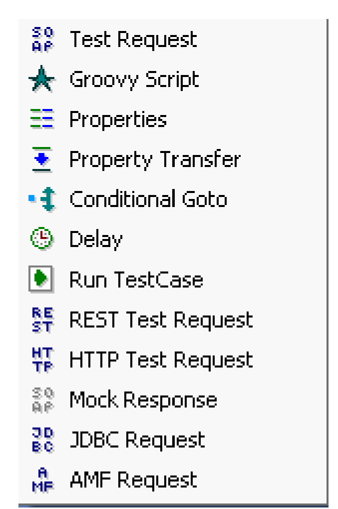
Cada TestCase en un TestSuite muestra un número de propiedades que pueden ser leídas, escritas o

modificadas por otros TestSteps, por ejemplo un paso de un script Groovy puede leer la propiedad de respuesta de un paso de petición y tomar alguna acción en función de su valor, véase la Propiedad de Expansión para más detalles y ejemplos.

Tipos de TestSteps

Actualmente están disponibles los siguientes TestSteps:

Tipo de Paso	Descripción
Request	Envía una petición SOAP y permite que la respuesta sea validada usando una variedad de aserciones
Property Transfer	Utilizadas para transferir los valores de propiedad entre dos TestSteps
Groovy Script	Ejecuta un script Groovy que puede hacer más o menos "cualquier cosa"
Properties	Se utiliza para definir propiedades globales que pueden ser leídas desde una fuente externa.
Conditional Goto	Permite cualquier número de saltos condicionales en la ruta de ejecución del TestCase. Las condiciones se especifican como expresión XPath y se aplican a la petición anterior de pasos de respuesta.
Delay Step	Pausa la ejecución de una TestCase durante un número especificado de milisegundos.
Run TestCase Step	Ejecuta otro TestCase dentro de uno ya existente
Rest Test Request	Ejecuta una Request Rest definida en el proyecto
HTTP Test Request	Ejecuta una llamada http
Mock Response	Escucha un respuesta SOAP que se valida y devuelve una respuesta mock
JDBC Request	Ejecuta una petición a una BBDD
AMF request	Ejecuta una petición AMF

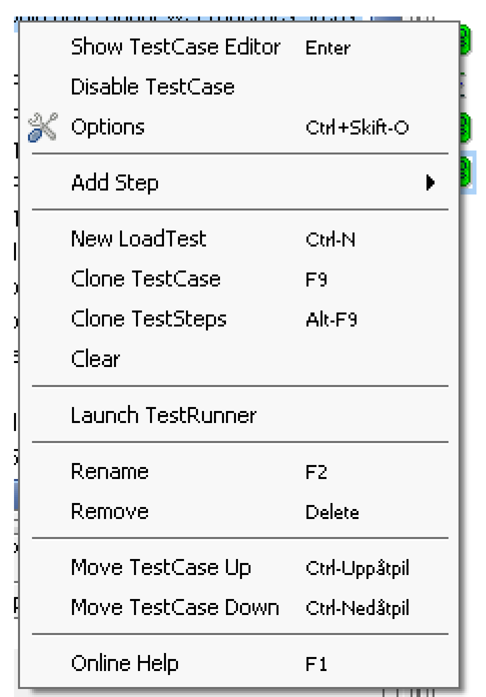


31 Tipos de TestSteps

Acciones de los TestCase

Las siguientes acciones están disponibles desde el menú que aparece al hacer click derecho en el nodo del TestCase:

- **Show TestCase Editor**: abre el editor del TestCase.
- **Disable/Enable TestCase**: deshabilita o habilita el TestCase.
- **Options**: muestra la ventana de opciones del TestCase.
- **Add Step**: añade un TestStep al TestCase.
- **New LoadTest**: abre una ventana para crear una nueva prueba de carga para el TestCase.
- **Clone TestCase**: abre una ventana para clonar todo el TestCase, opcionalmente en otro TestSuite.
- **Clear**: muestra una ventana para eliminar todos los TestSteps del TestCase.
- **Rename**: abre una ventana para renombrar el TestCase.
- **Remove**: muestra una ventana para eliminar el TestCase de su TestSuite.
- **Launch Runner**: abre el cuadro de diálogo de lanzamiento para el lanzador del TestCase de soapUI.
- **Move TestCase Up**: mueve el actual TestCase hacia arriba en la lista de los TestCase.
- **Move TestCase Down**: mueve el actual TestCase hacia abajo en la lista de los TestCase.
- **Online Help**: abre la página de ayuda online en el navegador.



32 Menú contextual nodo TestCase

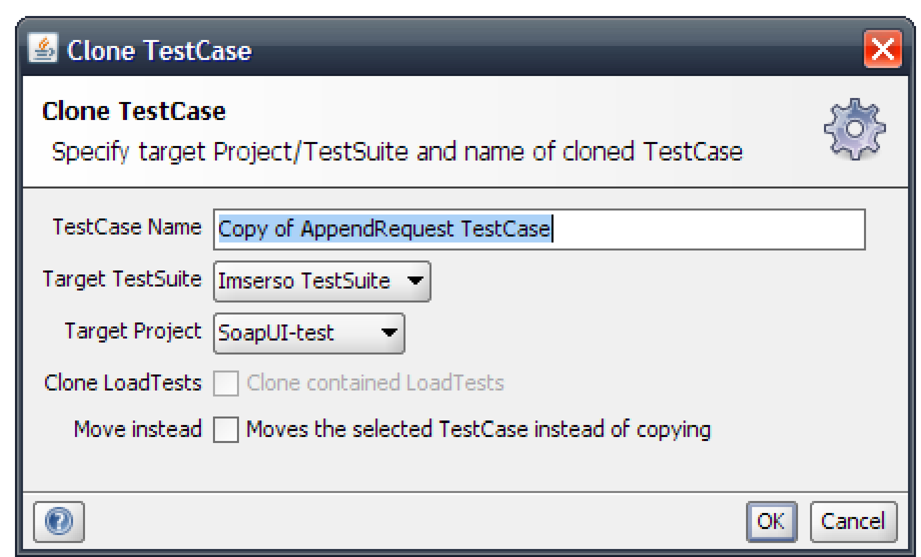
Etiqueta de detalle del TestCase

La pestaña de "detalles" que está en la esquina inferior izquierda, muestra los siguientes valores cuando el TestCase está seleccionado en el árbol de navegación

- **Name**: el nombre del TestCase actual.

Clonar un TestCase

Al seleccionar "Clone TestCase" del menú del TestCase abre el siguiente cuadro de diálogo:



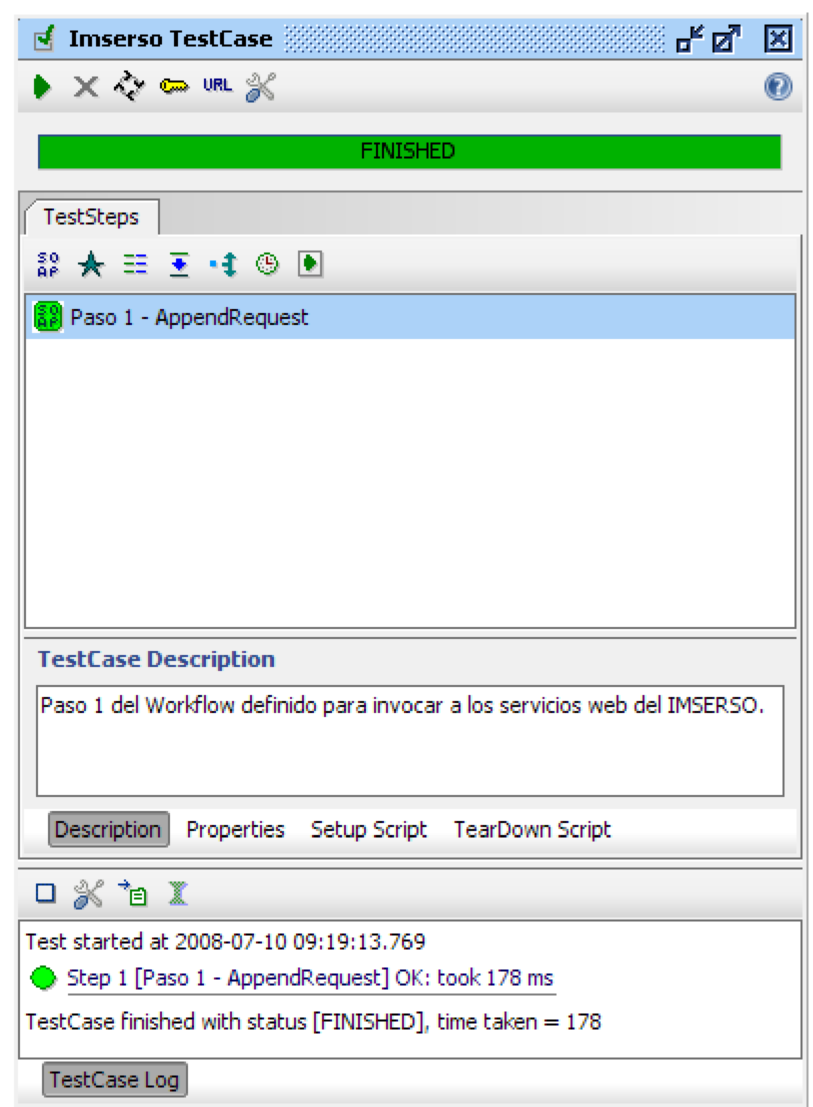
33 Clonar TestCase

Si seleccionas clonarlo a otro proyecto, soapUI abre una ventana para clonar las interfaces requeridas a ese proyecto.

El editor de TestCase

Haciendo doble click en el nodo del TestCase del navegador o seleccionando “Show TestCase Editor” de su menú de opciones abre el editor de TestCase, que permite editar y ejecutar el TestCase. Si el TestCase está haciendo una prueba de carga, el editor estará en su mayor parte deshabilitado. El editor está dividido en 4 partes (de arriba abajo):

- Una barra de herramientas para ejecutar o cancelar y las opciones de configuración
- Una barra de progreso que muestra el estado y el progreso del TestCase
- Pestaña TestSteps: la lista de TestSteps del TestCase
- Una serie de inspectores para el TestCase (de izquierda a derecha):
  - Description: una descripción parcial para el TestCase
  - Properties: propiedades del TestCase
  - Setup Script: un Groovy script para ejecutar cuando se ejecuta el TestCase
  - TearDown Script: un Groovy script para ejecutar cuando finaliza la ejecución del TestCase
- El registro del TestCase para ver y exportar los resultados. Se muestran las siguientes entradas:
  - Cuándo empieza el TestCase
  - Una entrada para cada TestStep ejecutado, indicando cuánto ha tardado
  - Errores y/o mensajes opcionales reportados por cada TestStep
  - Cuánto tiempo tardó el TestCase



34 Lanzando un test Case

La barra de herramientas del editor del TestCase

La barra de herramientas principal contiene las siguientes acciones (de izquierda a derecha)

- **Run TestCase**: ejecuta el TestCase.
- **Cancel TestCase**: cancela la ejecución de un TestCase.
- **Run Continuously**: cuando se selecciona el TestCase se ejecuta continuamente. Para detener el proceso hay que pulsar el botón de cancel del TestCase.
- **TestCase Credentials**: abre una ventana para configurar los credenciales que se van a usar en todas las peticiones del TestCase. Es bastante útil si quieres ejecutar tus pruebas con diferentes credenciales.
- **TestCase Endpoint**: abre una ventana para configurar el punto final para usar en todas las peticiones del TestCase. Es útil si quieres ejecutar tus pruebas contra diferentes servidores, etc.

Las URLs disponibles son recogidas en las interfaces de peticiones de operación.

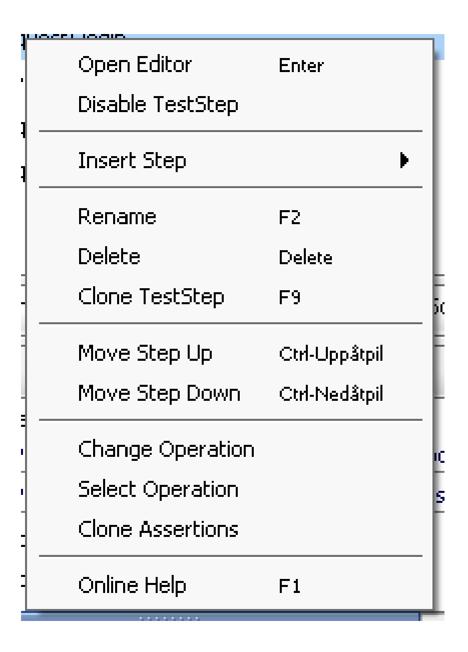
- **TestCase Options**: abre el cuadro de diálogo de opciones del TestCase descrito más adelante.
- **Online Help**: abre la página de ayuda online en el navegador.

Lista de TestSteps

La pestaña de "TestSteps" contiene una lista de los TestSteps configurados actualmente para este TestCase. Al hacer doble click en un TestStep de la lista, abre la ventana de edición de ese elemento. Al realizar botón derecho sobre un TestStep muestra un menú emergente con las siguientes acciones:

- **Open Editor**: abre el editor de TestSteps asociado (si está disponible)
- **Disable/Enable TestStep**: inhabilita o permite la ejecución del TestStep.
- **Insert Step**: muestra una lista de los TestSteps que se pueden insertar en la posición actual
- **Rename**: abre una ventana para renombrar el paso creado.
- **Delete**: abre una ventana para borrar el paso seleccionado.
- **Clone TestStep**: muestra una ventana para clonar el paso seleccionado.
- **Move Step Up**: mueve el paso seleccionado una posición hacia arriba en la lista (también se puede realizar con ctrl-flecha arriba).
- **Move Step Down**: mueve el paso seleccionado una posición hacia abajo en la lista (también se puede realizar con ctrl-flecha abajo).
- **Step specific actions**: depende de qué TestCase esté seleccionado.
- **Online Help**: abre la página de ayuda online en el navegador
- **Append Step**: muestra una lista de tipos de TestStep agregados para un TestCase

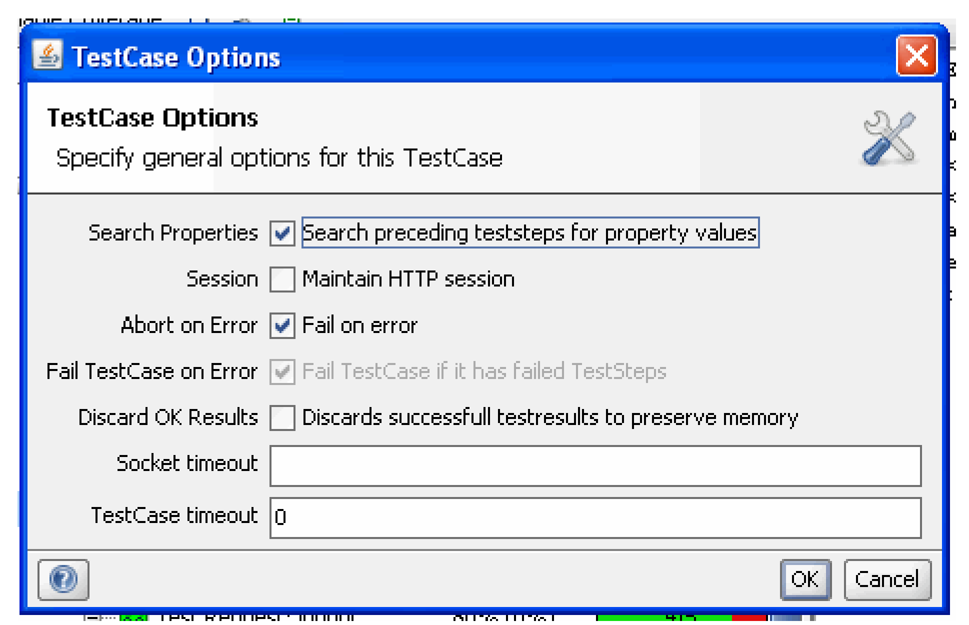
Haciendo doble click en la entrada de un TestStep del registro, abre el visor de resultados del TestStep seleccionado si está disponible y descrito en la página de documentación de cada TestStep (por ejemplo, el visor de resultados de la petición).



Opciones del TestCase

Al seleccionar las opciones del TestCase del nodo de TestCase del menú en el navegador ó de la barra de herramientas del editor del TestCase abre un cuadró de diálogo con las siguientes opciones.

- **Search Properties**: cuando se buscan los valores de la propiedad sin las especificaciones del paso, comprueba todos los pasos antes que el actual para la propiedad nombre.
- **Session**: controla que una sesión HTTP es mantenida para todas las peticiones en el TestCase. La elección de esto reutilizará cookies, autenticación de cabeceras, etc.
- **Abort on error**: controla si el TestCase tiene que ser cancelado cuando un TestStep falla con un error (por ejemplo, si una petición de un paso tiene aserciones fallidas).
- **Fail TestCase on Error:** controla si el TestCase va a fallar, si la opción de “Fail on error” no está seleccionada y el TestCase termina con uno o más TestSteps.
- **Discard OK Results**: la larga ejecución de los TestCase eventualmente consume gran cantidad de memoria ya que los resultados de los TestSteps están internamente cacheados para su posterior visualización y presentación. Al seleccionar esta opción, hará que soapUI sólo guarde los resultados sin éxito de los TestSteps con lo que se ahorrará importantes cantidades de memoria.
- **Socket timeout**: el timeout (en milisegundos) que se utilizará para todas las peticiones del TestCase.
- **TestCase timeout**: el timeout (en milisegundos) a esperar antes cancelar o que falle la ejecución de un TestCase.



35 Opciones de un TestCase

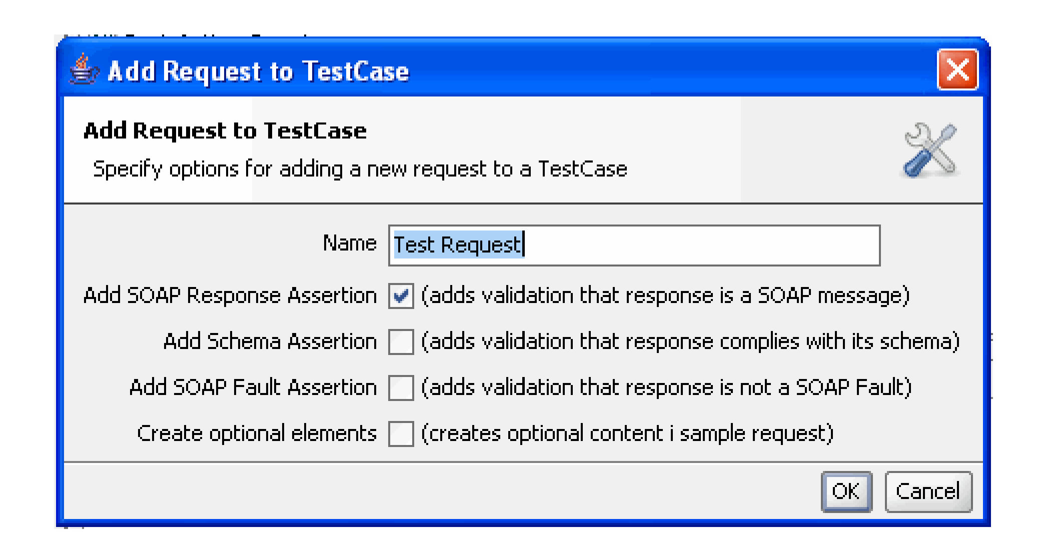
4.3.1. Test Requests

Los TestRequests son una de las principales características cuando se trabaja con soapUI. Extiende peticiones estándar con la posibilidad de añadir cualquier número de aserciones que se aplicarán a la respuesta recibida por la petición. Esto es, comprueba que la respuesta contenga lo que se espera que contenga.

Para crear un TestRequest se puede hacer de 2 maneras. Desde las peticiones estándar usando la acción de "Add to TestCase" o a través del menú emergente del editor de TestCase con la opción "Add Step ->

Test Request", el cuál mostrará una ventana en la que hay que indicar para qué interface u operación tiene que ser creada la petición.

En ambos casos, se mostrará un cuadro de diálogo para añadir aserciones estándar para hacer la prueba del servicio web más rápida.

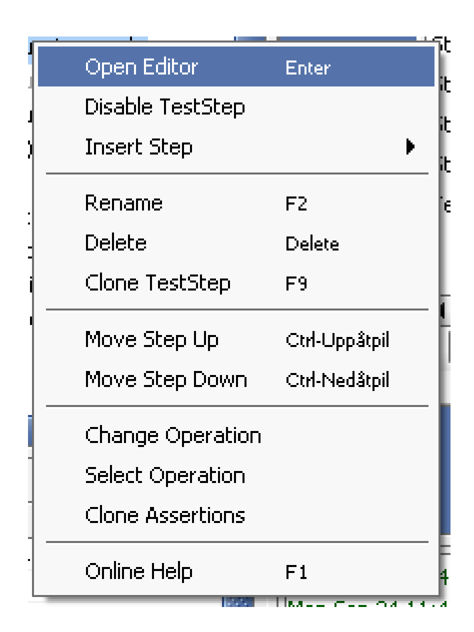


Los TestRequests son enviados manualmente a través de las acciones de envío de sus editores o cuando se ejecuta el TestCase que contiene la petición. La respuesta de la petición es validada contra las aserciones de peticiones y el icono de la petición cambia para reflejar el resultado de la validación; verde significa que todas las validaciones fueron bien y rojo que algunas fallaron. Un icono con el fondo gris indica que la petición todavía no ha sido enviada para validar, un fondo blanco indica que los TestRequests carecen de aserciones.

Acciones de los TestRequests

Las siguientes acciones están disponibles al hacer click derecho sobre el nodo del TestRequest del menú:

- **Open Editor**: abre el editor del TestStep asociado (si está disponible).
- **Disable/Enable TestStep**: inhabilita o permite la ejecución del TestStep.
- **Insert Step**: muestra una lista de los TestStep que se pueden insertar en la posición actual.
- **Rename**: abre una ventana para renombrar el paso seleccionado.
- **Delete**: abre una ventana para borrar el paso seleccionado.
- **Clone TestStep**: muestra una ventana para clonar el paso seleccionado.
- **Move Step Up**: mueve el paso seleccionado una posición hacia arriba en la lista (también se puede realizar con ctrl-flecha arriba).
- **Move Step Down**: mueve el paso seleccionado una posición hacia abajo en la lista (también se puede realizar con ctrl-flecha abajo).
- **Change Operation**: abre una ventana para modificar la operación del TestRequest.
- **Select Operation**: selecciona este TestRequest en el navegador.
- **Online Help**: abre la página de ayuda online en el navegador.



Detalles de la pestaña del TestRequest

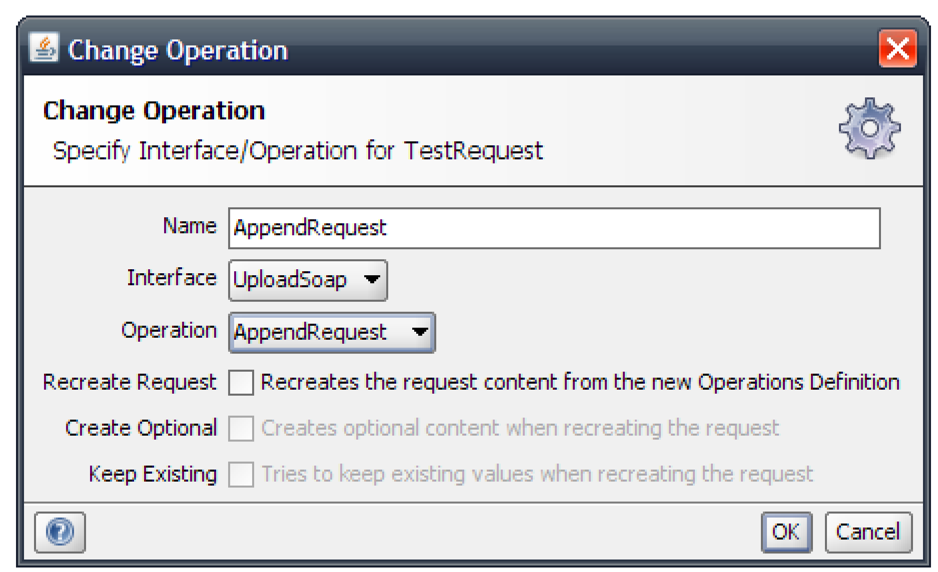
La pestaña de "detalles" situada en la esquina inferior izquierda muestra las mismas propiedades que cuando un nodo de TestRequest está seleccionado en el árbol de navegación para una petición estándar. Añade dos propiedades de sólo lectura:

- **Interface**: el nombre de la interface para este TestRequest
- **Operation**: el nombre de la operación para este TestRequest



Cambiar operación

Al seleccionar "change operation" de un TestRequest abre el siguiente cuadro de diálogo:



36 Ventana "Change Operation"

La lista de interfaces muestras las interfaces disponibles para el proyecto actual. La lista de operaciones se actualiza respectivamente.

El editor de la TestRequest

Al hacer doble click en un TestRequest, tanto en el navegador como en el cuadro del editor del TestCase, abre el editor de petición, el cuál es más o menos una copia del editor de petición estándar con las siguientes excepciones:

- El segundo botón de la barra de herramientas “Add to TestCase” ha sido reemplazado por “Add Assertion”, el cuál muestra una ventana para añadir una aserción al TestRequest.
- La acción de clonar ahora clona el TestRequest y adjunta la petición clonada al contenedor de TestCase
- Debajo del cuadro petición/respuesta hay un cuadro que contiene 2 nuevas pestañas; la pestaña “Assertions” (aserciones) y la pestaña “Request Log” (registro de peticiones).

Todas las demás funcionalidades de edición, presentación y validación son las mismas que en el editor de petición.

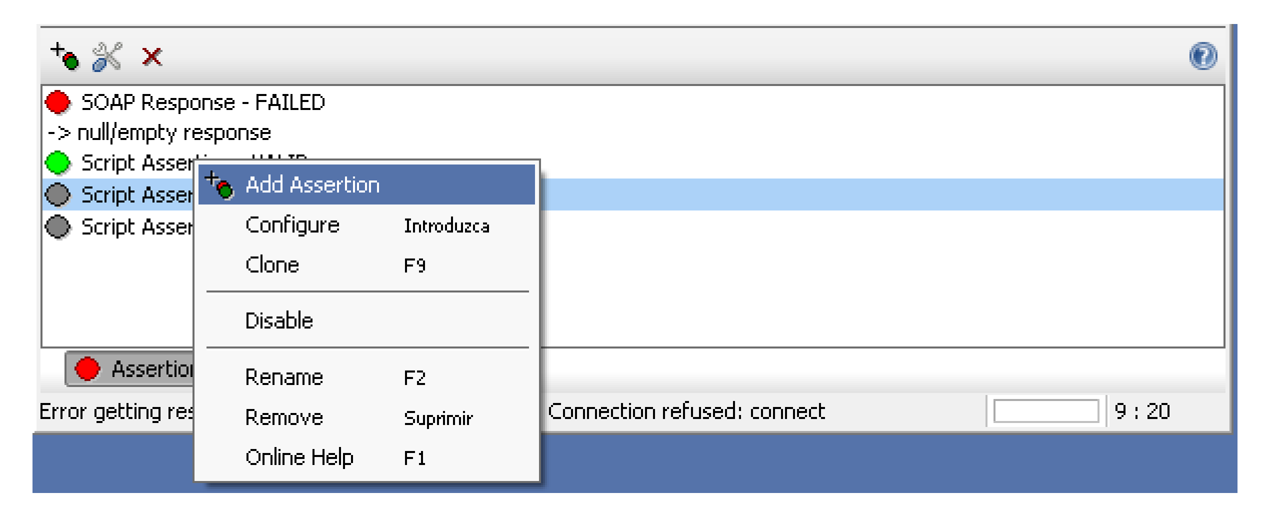
La pestaña de aserciones

La pestaña de aserciones lista las aserciones que han sido configuradas para el TestRequest. Haciendo doble click en una aserción de la lista abre el cuadro de diálogo de configuración de aserciones (si está disponible). Se pueden añadir tantas aserciones como sean necesarias y a veces puede ser útil añadir el mismo tipo de aserciones varias veces con diferentes configuraciones.

Un círculo pintado al lado de la aserción indica el estado de ésta de acuerdo a la última respuesta recibida; rojo = la aserción ha fallado junto con los mensajes de error, verde = aserción ok, gris = la aserción no ha sido ejecutada.

Las siguientes acciones están disponibles al hacer click con el botón derecho en el menú de la lista de aserciones:

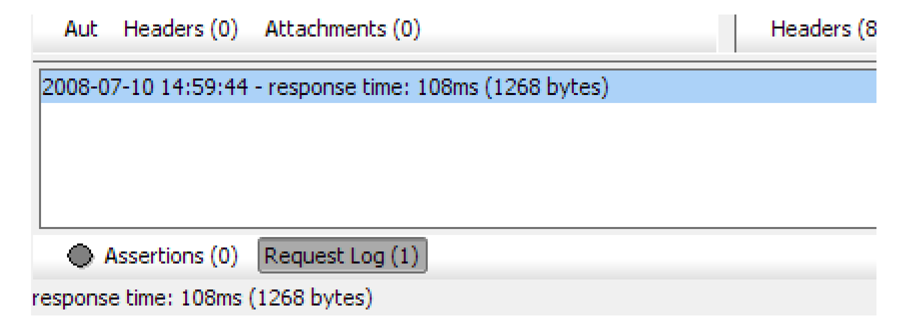
- **Add Assertion**: muestra una ventana para añadir una nueva aserción a la lista.
- **Configure** (si es posible): abre el cuadro de diálogo de configuración de la aserción seleccionada.
- **Clone** (si es posible): abre una ventana para clonar la aserción seleccionada.
- **Disable**: habilita/deshabilita las aserciones seleccionada.
- **Rename**: muestra una ventana para renombrar la aserción seleccionada.
- **Remove**: abre una ventana para borrar la aserción seleccionada.



La barra de herramientas contiene los botones de añadir, configurar y borrar una aserción.

La pestaña de registro de petición

La pestaña de registro de petición muestra la fecha y hora de envío, la duración y el tamaño de la respuesta para un TestRequest. Esto puede ser práctico si quieres comparar manualmente los tiempos de respuesta y el tamaño.



Aserciones de respuesta

Una de las principales características de soapUI es la capacidad de crear aserciones sobre el contenido de las respuestas SOAP. Se suministran una serie de diferentes aserciones con el fin de cubrir las necesidades de calidad para pruebas. Actualmente las siguientes aserciones están disponibles.



Tipo	Descripción

Schema Compliance *	Valida el mensaje de respuesta contra su esquema xml
Simple Contains	Comprueba la existencia de un símbolo
Simple Not Contains	Comprueba la no existencia de un símbolo
SOAP Fault *	Comprueba que la respuesta es un fallo soap
Not SOAP Fault *	Comprueba que la respuesta no es un fallo soap
SOAP Response *	Comprueba que la respuesta es una respuesta SOAP válida
Response SLA *	Comprueba que el tiempo de respuesta sea menor que uno especificado
XPath Match	Comprueba si una determinada expresión XPath coincide con un valor predefinido
XQuery Match	Comprueba si una determinada expresión XQuery coincide con un valor predefinido
Script Assertion	Permite el uso de un script Groovy para hacer valer el intercambio de mensajes
WS-Security Status *	Comprueba que el procesamiento de WS-Security sea correcto

Las aserciones marcadas con (*) son "aserciones únicas", lo que indica que sólo pueden ser añadidas una sola vez al TestRequest.

Aserción de conformidad de esquema

La aserción de conformidad de esquema comprueba que la respuesta es compatible con el esquema xml de los mensajes. Si no lo es, una lista de errores de validación, como aquellos mostrados en la pestaña de validación del editor de petición, son mostrados y la aserción falla. La lista de errores también se muestra en la pestaña de aserción del editor de TestRequest; si es posible, se puede hacer doble click sobre los errores para resaltar la línea del error de validación.

La aserción tiene un único parámetro de configuración que es mostrado cuando se crea/configura una aserción de conformidad de esquema, la URL para la definición WSDL que se usa para validación. Por defecto es la URL de la definición que contiene la interface de operaciones del TestRequest.

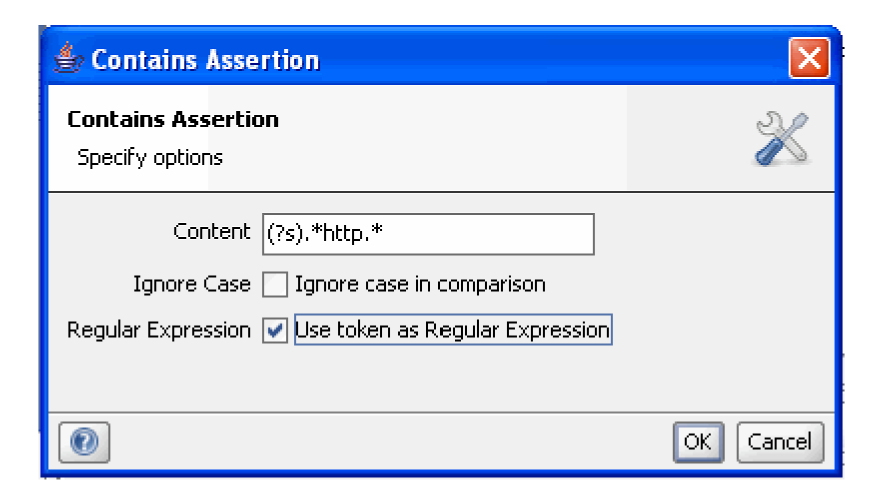
La aserción de conformidad de esquema está dirigida a un perfil básico compatible con los mensajes WSDL y, por lo tanto, solo admite mensajes literales codificados (rpc y document). La validación de mensajes SOAP-encoded no se admite y se traducirá en un error de validación.

Hay que tener en cuenta el hecho que un SOAP-Fault sólo será un esquema de validación si el elemento de detalle contiene parte del mensaje que está definido en el correspondiente enlace y que no será compatible con su correspondiente esquema de comunicación (ver, más adelante, la aserción de SOAP-Fault en relación

con las validaciones de mensajes de respuesta de SOAP-Fault).

Simple Contains

La aserción "Simple Contains" chequea la existencia de un substring específico en la respuesta recibida. En todos los casos no se realiza análisis sintáctico ni validación.
La aserción tiene 3 parámetros de configuración que son mostrados en una ventana cuando se crea o configura un contenedor simple de aserción:



Las opciones son:

- **Content**: el contenido a buscar.
- **Ignore Case**: Ignora diferencias entre mayúscula y minúscula.
- **Regular Expresión**: evalúa como expresión regular.

Simple Not Contains

La aserción "Simple NotContains" chequea la no existencia de un substring específico en la respuesta recibida. En todos los casos no se realiza análisis sintáctico ni validación.
La aserción tiene los mismos parámetros de configuración descritos para el contenedor de aserciones de arriba.

SOAP Fault

La aserción “SOAP Fault” comprueba que la respuesta recibida es un fallo soap. Esta aserción no tiene parámetros de configuración.

Not SOAP Fault

La aserción “Not SOAP Fault” comprueba que la respuesta recibida no es un fallo soap. Esto tiene que ser usado junto al conformidad de esquema ya que un SOAP fault no se valida contra ningún esquema (al menos hay una parte del fallo definido en el WSDL y ese fallo está presente en la respuesta). Esta aserción no tiene parámetros de configuración.

SOAP Response

Valida que la respuesta es un mensaje válido de SOAP. Esta es la mínima aserción que debería ser añadida para recoger respuestas vacías o páginas de error de HTTP. Esta aserción no tiene parámetros de configuración.

Response SLA

La aserción “Response SLA” comprueba que el tiempo de respuesta es menor que el tiempo límite indicado.

Aserción WS-Security Status

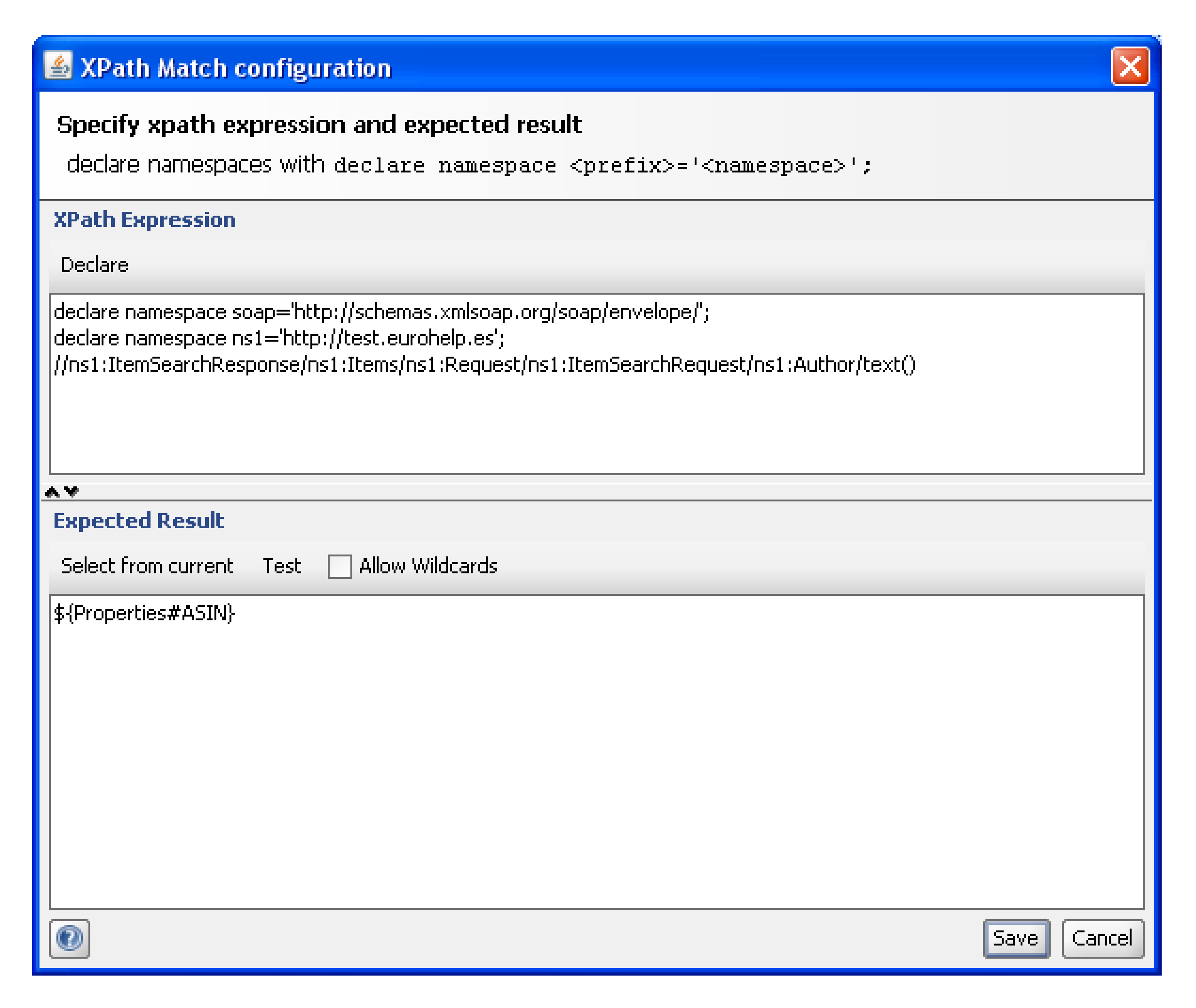
La aserción “WS-Security Status” comprueba que el mensaje de entrada tiene cabeceras válidas de WS-Security.

XPath Match

La aserción “XPath Match” permite la especificación de una expresión XPath para ser evaluada en relación a un mensaje de respuesta recibido y comparar su resultado con un valor predefinido. Las expresiones pueden seleccionar todo, desde valores de atributos, hacer evaluaciones booleanas o seleccionar la respuesta de todo el cuerpo (XmlUnit se usa internamente para comparar los elementos XML, jerarquías o nodos).

Internamente, soapUI usa el motor Saxon 8.8 XPath que tiene soporte para XPath 1,0 y 2,0 XPath.

El diálogo de configuración para la aserción XPath Match se divide en 2 zonas, la de la parte superior que contiene la expresión XPath deseada y la de la parte inferior que contiene el resultado esperado.



La barra de herramientas para la parte superior del XPath contiene la siguiente acción:

- Declare: Añade una declaración en el espacio de nombre para todos los espacios de nombres que actualmente se definen en el mensaje subyacente de respuesta a la expresión XPath. Saxon utiliza esta sintaxis para las declaraciones de espacios de nombres para que se puedan utilizar más tarde en la expresión XPath.

La barra de herramientas para la parte inferior contiene las siguientes acciones (de izquierda a derecha):

- Select from current: evalúa la expresión XPath especificada contra el actual mensaje de respuesta (si está disponible) para la petición subyacente. El resultado se escribe en el área de resultados del diálogo de configuración
- Test: evalúa la expresión XPath especificada contra el actual mensaje de respuesta (si está disponible) para la petición subyacente y compara este resultado con el valor especificado en el área de resultados. Esta es esencialmente la misma comparación que se realiza durante una aserción "verdadera".
- Allow Wildcard: Permite el uso del comodín '*' en atributos y elementos de valores. Éstos se evitarán durante la comparación (véase el tutorial más adelante).

La barra de botones de abajo muestra el siguiente:

- Help: abre la página de ayuda online en el navegador

- Save: guarda los valores resultados del actual XPath y cierra el cuadro de diálogo.
- Cancel: descarta los valores resultados del actual XPath y cierra el cuadro de diálogo.

El diálogo es de ventana no modal, lo que te permite pasar a centrarse de nuevo en soapUI y, por ejemplo, seleccionar los valores subyacentes del mensaje de respuesta.

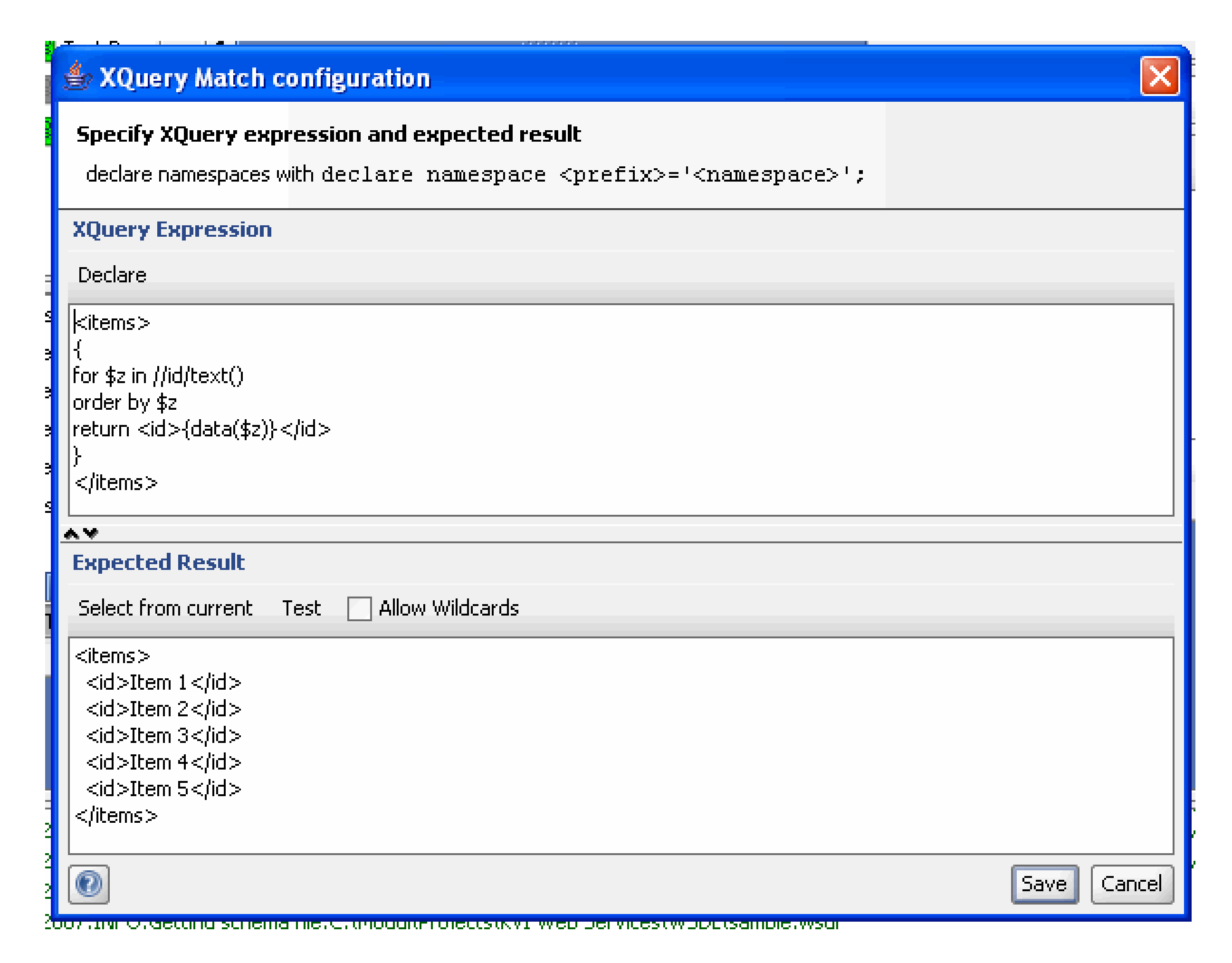
Creando aserciones XPath

El método recomendado para crear una aserción XPath es la siguiente:

1. Enviar el TestRequest y esperar su respuesta (por lo tanto hay una respuesta para probar)
2. Crear la aserción de XPath y comenzar añadiendo las declaraciones de espacios de nombres a la expresión XPath con el botón “declare”
3. Ahora se añade la expresión XPath después de las declaraciones de los espacios de nombres, count (//ns1:Item) en la captura de pantalla anterior
4. Utilice el botón "Select from current " para evaluar la expresión XPath contra el resultado disponible y comprobar que devuelve lo que se espera, 10 en la captura de pantalla anterior
5. Vuelve a comprobar pulsando la tecla "Test" que debería devolver un mensaje de OK ya que la prueba comparará el resultado de la selección XPath con el valor seleccionado previamente.

Aserción XQuery Match

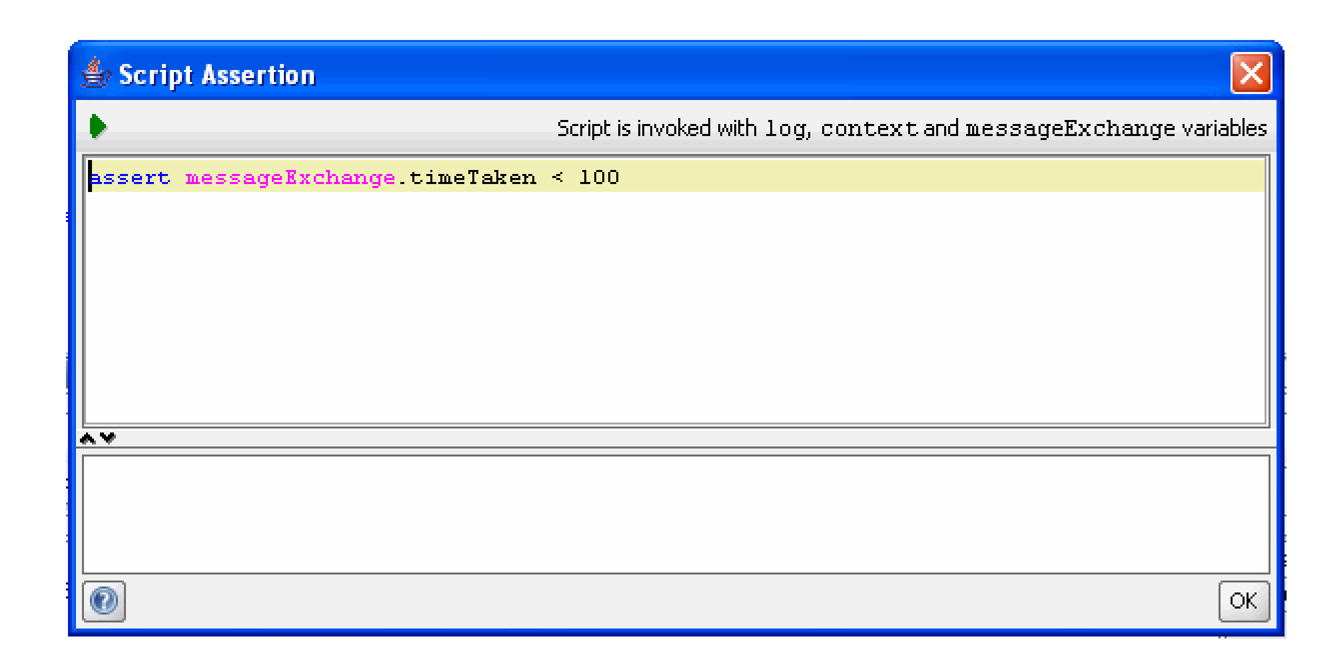
La aserción XQuery Match está configurada exactamente como la aserción XPath Math que se ha descrito anteriormente, la única diferencia es que la expresión se evalúa como una expresión XQuery 2.0.



Esta aserción es útil cuando se quiere asegurar un subconjunto de los datos y, por ejemplo, no depender de otros datos. En el ejemplo de la captura de pantalla, una lista ordenada de ids se crea a partir de la respuesta y es comparada contra una lista predefinida, lo que resulta en una aserción que no fallará si los elementos vienen en otro orden y/o obtienen más/otros datos "en torno a" la id del elemento.

Aserción Script

La aserción “script” permite validaciones arbitrarias (ver los ejemplos de abajo). Al crear o al hacer doble click en una aserción “script”, un editor de script Groovy se muestra como se ve a continuación



La aserción “script” se puede ejecutar en contra del último intercambio de mensaje con el botón de ejecución en la parte superior izquierda.

El script tiene acceso a los siguientes objetos:

- messageExchange: el messageExchange para la actual petición / respuesta. Da acceso directo al contenido de los mensajes, cabeceras HTTP, archivos adjuntos, etc.
- context: el TestRunContext ejecuta el actual TestCase, que a partir de ahora será una instancia de WsdlTestRunContext
- log: un objeto estándar de log4j Logger disponible para el registro arbitrario de información

Una aserción “script” debería lanzar una excepción con el mensaje de error para fallar la aserción. Se puede utilizar también una declaración construida en Groovy para una fácil sintaxis de la aserción, tal como se muestra en los ejemplos de a continuación. Si la aserción es válida, devolverá nada o un mensaje de estado que se muestra en el registro del TestCase.

Ejemplos de aserción script

Validar un cierto tiempo de respuesta:

```
assert messageExchange.timeTaken < 400
```

Validar la existencia de una cabecera de respuesta HTTP:

```
assert messageExchange.responseHeaders["x-amz-id-1"] != null
```

Validar la existencia de un elemento específico utilizando GroovyUtils (aunque esto sería más fácil con una aserción estándar XPath- Contains):

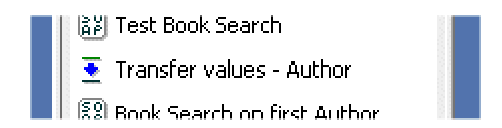
```
def groovyUtils = new com.eviware.soapui.support.GroovyUtils( context )
def holder = groovyUtils.getXmlHolder( messageExchange.responseContent )

assert holder["//ns1:RequestId"] != null
```

4.3.2. Property Transfers

La Property-Transferes son TestStep que transfieren propiedades entre los contenedores de propiedad dentro del mismo alcance que la Property-Transfer del TestStep (es decir, que contenga su TestCase, su TestSuite, el proyecto y las propiedades del destino). El paso puede contener un número arbitrario de "transferencias" especificando una propiedad de origen y destino con expresiones opcionales de XPath / XQuery.

Las Property-Transferes utilizan el mismo motor de Saxon XPath / XQuery descrito para las aserciones XPath y XQuery.

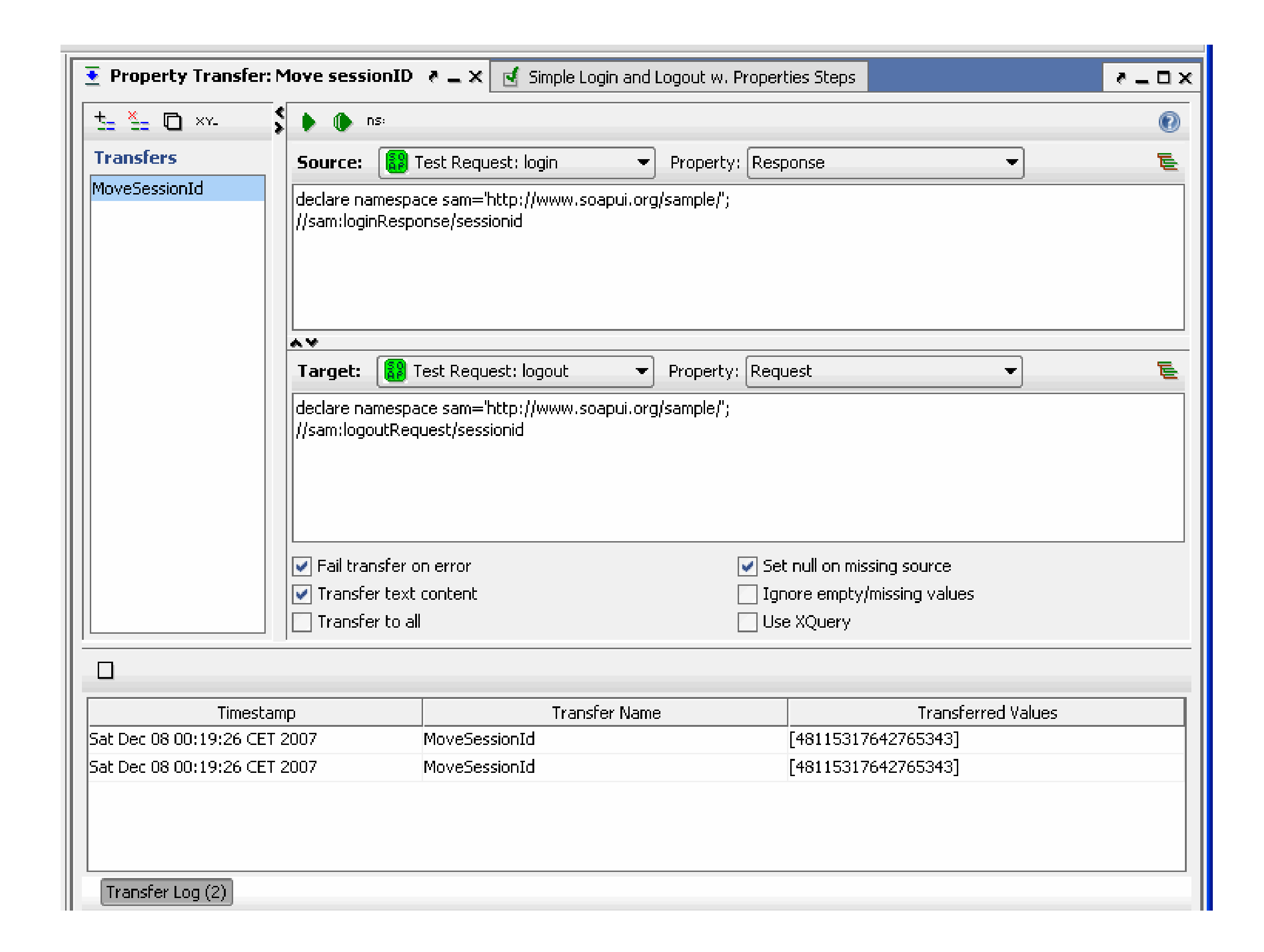


El editor de Property-Transferes

El editor de Property-Transferes se abre haciendo doble click en un paso de una Property-Transfer del navegador o en la lista de TestStep del editor de Testcase.

El editor contiene una lista de transferencias configuradas a la izquierda, seleccionando una transferencia de la lista mostrará su código y la expresión de destino de XPath y XQuery a la derecha.

La lista desplegable de la parte superior derecha se utiliza para especificar el origen y la propiedad a ser transferida. Los destinos y propiedades se especifican con la lista desplegable del medio. Si las propiedades contienen un XML seguido de una expresión XPath se pueden especificar aún más, seleccionando el valor a transferir o mover.



Las siguientes botones están disponibles en la barra de herramientas superior:

- **Add**: muestra una ventana para añadir una nueva transferencia a la lista.
- **Delete**: abre una ventana para borrar la transferencia seleccionada.
- **Copy**: abre una ventana para crear un copia de la transferencia seleccionada.
- **Rename**: muestra una ventana para renombrar la transferencia seleccionada.
- **Run**: ejecuta la transferencia seleccionada, es decir, transfiere el valor según la configuración realizada. Para que funcione correctamente el origen y la propiedad destino tienen que estar disponibles (por ejemplo una respuesta de un TestRequest).
- **Run all**: ejecuta todas las transferencias, es decir, transfiere los valores especificados. Para que funcione correctamente el origen y la propiedad destino tienen que estar disponibles (por ejemplo una respuesta de un TestRequest).
- **Declare**: declara el espacio de nombres de los campos XPath de origen y destino. El espacio de nombres del campo origen se extrae del mensaje de respuesta del TestRequest seleccionado. El espacios de nombre del campo destino es extraído de los correspondientes mensajes de petición de los TestRequests. Si cualquiera de estos no están disponibles, soapUI muestra una ventana para definir todos los nombres disponibles de los esquemas asociados.
- **Online Help**: abre la página de ayuda online en el navegador.

El registro de transferencia de la parte inferior del editor muestra todas las transferencia realizadas por la Property-Transfer mientras el editor esté abierto, incluyendo tanto las realizadas usando los botones de la barra

de herramientas y las realizadas durante la ejecución de un TestCase/TestSuite. La barra de herramientas del registro de transferencia contiene un botón "clear" para borrar su contenido actual.

Ejecutando una transferencia

Tras la ejecución de un TestCase, cada transferencia de la Property-Transfer se realiza mediante la selección de las propiedades especificadas por el paso de origen, por la propiedad y expresión XPath opcional y copiando su valor a la propiedad especificada por el paso de destino pudiendo usar una expresión XPath opcional.

Si las expresiones XPath están especificadas, soapUI tratará de sustituir el nodo destino con el nodo de origen si son del mismo tipo. Si no (por ejemplo, cuando se asigna texto () a un @atributo), soapUI hará todo lo posible para copiar el valor.

Las expresiones XPath de origen y destino deben apuntar a nodos existentes en sus respectivas propiedades. La propiedad origen, obviamente, requiere el nodo para que pueda ser seleccionado y la propiedad destino requiere el nodo para que pueda ser encontrado y sobrescrito.

Si alguna de las transferencias falla debido a que falta alguna expresión XPath, se mostrará un error y el paso puede fallar o continuar, dependiendo si la opción “Fail on error” ha sido seleccionado para dicha transferencia. La ejecución de un TestCase sólo se aborta si la opción “Fail on error” del TestCase ha sido seleccionada, tal como se describe en las opciones del TestCase.

Las siguientes opciones están disponibles para cada transferencia:

- **Fail transfer on error**: falla la Property-Transfer si ocurre un error (por ejemplo la falta de una propiedad de origen).
- **Set null on missing source**: anula el error de la falta de valores de origen y fija las propiedad de destino a null en estos casos.
- **Transfer text content**: Cuando la expresión XPath apunta a nodos elementos, su contenido de texto se transfiere en lugar de los elementos por sí mismos (necesarios para la compatibilidad con versiones anteriores con soapUI 1.5).
- **Ignore empty/missing values**: anula los errores para la falta de valores de origen e ignora sus correspondientes transferencias.
- **Transfer to all**: Si la expresión de destino XPath selecciona múltiples nodos, el valor de la propiedad de origen se establecerá en todos estos nodos en lugar de sólo el primero.
- **Use XQuery**: interpreta la expresión de origen XPath especificada como una expresión XQuery, lo que permite la transferencia de datos complejos/transformados en lugar de sólo "simple" copiar.

Trabajando con Property-Transfer

Una Property-Transfer se puede crear de las siguientes maneras:

1. Primero hay que crear los 2 TestStep entre los que hay que realizar la transferencia
2. Crear una Property-Transfer en el diálogo de configuración usando el botón “add”
3. Selecciona los TestStep de origen y destino y su correspondientes propiedades de las listas desplegables.
4. Si alguna de las propiedades contiene XML, hay que proceder definiendo el espacio de nombres en las expresiones XPath utilizando el botón “define”. Luego hay que añadir las expresiones XPath que especifican qué seleccionar y dónde copiarlo. En la captura de pantalla de arriba, ambas expresiones son //ns1:SessionId, lo que indica que el elemento sessionID será copiado de la respuesta anterior a la solicitud siguiente (que debe estar disponible, pero preferiblemente vacía)
5. Prueba la transferencia seleccionando el botón “run” y comprueba que los valores de la siguiente petición se han copiado correctamente. Cualquier error será mostrado ya sea en el registro principal de soapUI (en la parte inferior) o en una ventana emergente.
6. Repite los pasos 3-5 para cada transferencia añadida al ValueTransfer

4.3.3. Conditional Gotos

Los pasos de condicional Goto evalúan un número arbitrario de condiciones XPath del mensaje de respuesta de la petición anterior y transfieren la ejecución del TestCase al TestStep asociado con la primera condición que se evalúe a verdadero. Esto permite la ejecución condicional de rutas de TestCase, donde el resultado de algunas peticiones controla cómo moverse por el TestCase. Si no hay condiciones que coincidan con la respuesta actual, la ejecución del TestCase continúa después del paso Goto de forma habitual.

Ejemplo de escenarios:

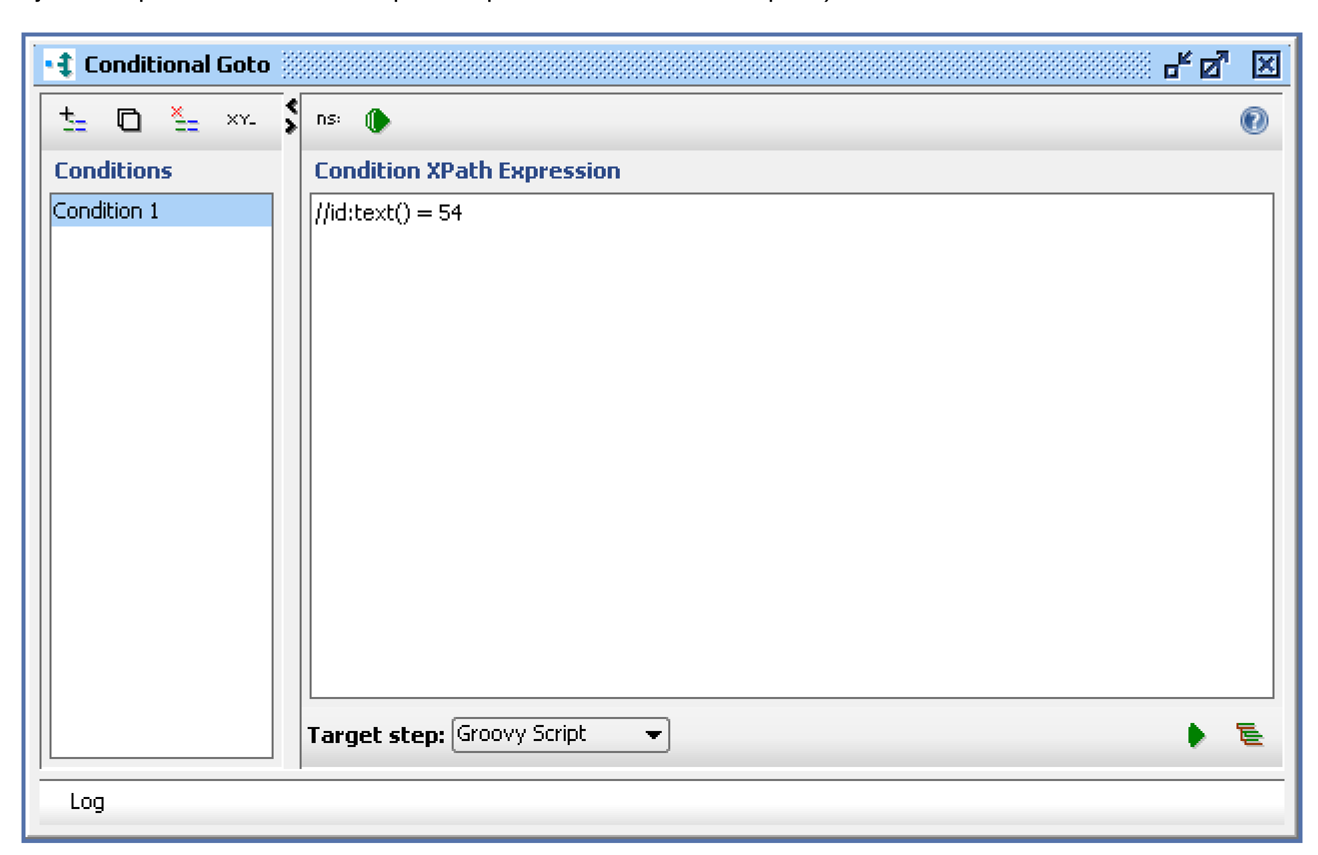
- Ramificaciones que dependen de los resultados devueltos por una petición
- Reiniciar después de un largo retraso en las pruebas de vigilancia
- En varias ocasiones esperar y chequear el valor de estado antes de continuar (por ejemplo en un proceso por lotes)

Las condiciones usan el mismo motor que Saxon XPath descrito para las aserciones XPath. Recuerda que una condición debe evaluar un valor booleano para ser válida (ver ejemplos más abajo).

El editor del condicional Goto

El editor del condicional Goto se abre haciendo doble click en un TestStep de un condicional Goto, tanto en el navegador como en la lista de TestStep del editor de TestCase.

El editor contiene una lista de condiciones configuradas a la izquierda, seleccionando una condición de la lista mostrará, a la derecha, la expresión de esa condición y una lista desplegable para el TestStep de destino. El botón de prueba de la condición evaluará la condición con la respuesta actual y mostrará el resultado (un mensaje de respuesta debe estar disponible para el anterior TestRequest).



Las siguientes acciones están disponibles en la barra de herramientas superior:

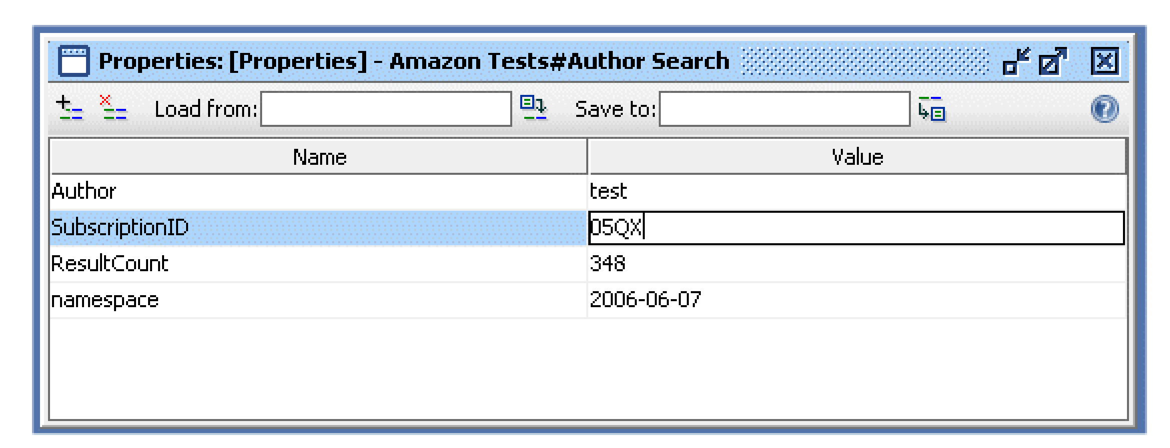
- **Add**: muestra una ventana para añadir una nueva condición a la lista.
- **Copy**: abre una ventana para crear un copia de la condición seleccionada.
- **Delete**: abre una ventana para borrar la condición seleccionada.
- **Rename**: muestra una ventana para renombrar la condición seleccionada.
- **Run**: ejecuta las condiciones y muestra la condición que coincide con la respuesta actual (un mensaje de respuesta debe estar disponible para el anterior TestRequest).
- **Declare**: declara el espacio de nombres en el campo expresión de la condición seleccionada. El espacio de nombres es extraído del actual mensaje de respuesta de la petición de respuesta precedente.
- **Online Help**: abre la página de ayuda online en el navegador.

4.3.4. Properties Step

Un *Property-Step* permite definir un número arbitrario de propiedades que se pueden usar desde las Property-Transfers y desde los pasos de un script Groovy. Las propiedades, opcionalmente, pueden ser leídas de y escritas en un fichero de propiedades bajo ejecución, por ejemplo, si se desea especificar algunas propiedades externas (contraseñas, puntos finales, etc) o si se quiere escribir algunos resultados en un fichero para su posterior presentación de informes.

Editor de propiedades

El editor de paso de propiedad es sencillo:



La barra de herramientas contiene lo siguiente (de izquierda a derecha):

- **Add Property**: abre una ventana para añadir una nueva propiedad.
- **Remove Property**: muestra una ventana para borrar la propiedad seleccionada
- **Load from**: campo opcional que contiene un archivo/URL/sistema de propiedad que contiene el código de las propiedades. El archivo especificado o la URL serán leídas como un fichero estándar de propiedades y el valor que tiene la propiedad se asignará a los pasos de propiedad. Si el campo está fijado para el nombre de un sistema de propiedad, esta propiedad debe especificar un fichero o una URL que será leído posteriormente, tal como se ha descrito
- **Select Properties Source**: permite la selección de un fichero local que contiene las propiedades

a ser leídas. El fichero seleccionado será leído y los valores de las propiedades contenidas serán asignados a las propiedades que coincidan en los pasos de propiedad (se mostrará una ventana si se intenta crear propiedades que no están disponibles)

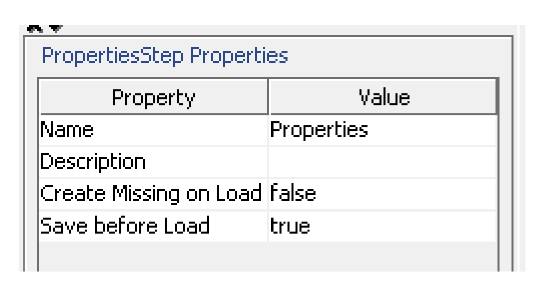
- **Save to**: campo opcional que contiene un fichero o un sistema de propiedad que contiene el nombre de destino del fichero de propiedades. El archivo indicado se creará como un fichero estándar de propiedades y los valores de propiedad que contiene serán escritos en el mismo. Si el campo está fijado para el nombre de un sistema de propiedad, esta propiedad debe especificar un archivo que será creado posteriormente, tal como se ha descrito.
- **Select Properties Target**: permite la selección de un fichero local en el que las propiedades deben estar escritas

El cuadro que aparece bajo la barra de herramientas muestra las propiedades definidas y sus valores, los valores y los nombres pueden ser modificados por la edición estándar.

Pestaña de detalles de propiedad

La pestaña de detalle del paso de propiedad (parte inferior izquierda) contiene lo siguiente:

- **Name**: el nombre del paso de propiedad.
- **Description**: una pequeña descripción del paso de propiedad.
- **Create Missing on Load**: crea las propiedades de los ficheros de propiedad de origen que actualmente no están definidos.
- **Save before Load**: se utiliza para salvar las propiedades existentes antes de cargar otras nuevas de los ficheros de propiedad de origen y destino.



Ejecución del paso

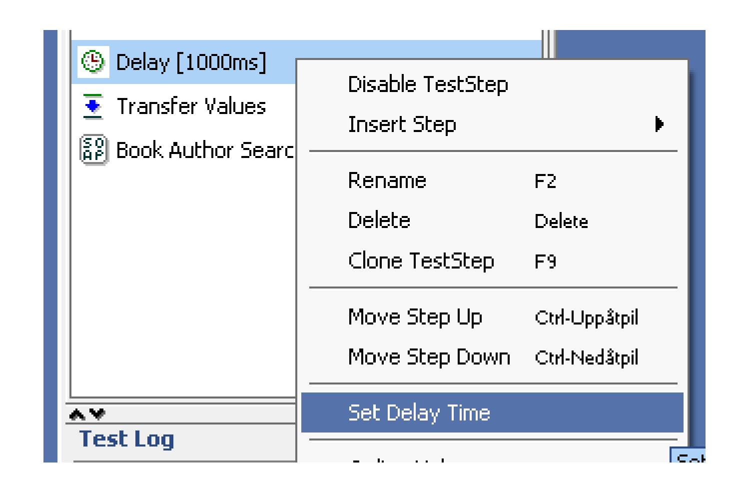
Cuando un paso de propiedad se ejecuta durante un TestCase, se toman las siguientes acciones:

- Las propiedades son leídos o escritos a partir de una fuente si se ha especificado como se describe más arriba (dependiendo de la opción " Save before Load ")
- Las propiedades se escriben o leen a un objetivo si se ha especificado como se describe más arriba (dependiendo de la opción " Save before Load ")
- Las propiedades son todas copiados a las actuales propiedades de TestRunContexts para que estén disponibles para la Propiedad de Expansión

4.3.5. Delay Step

Los TestStep de retardo se pueden insertar en cualquier posición en un TestCase para pausar la ejecución de un TestCase durante un número determinado de milisegundos

Para insertar un paso de retardo se utiliza la opción del menú “Insert Step” que se ha descrito anteriormente. Al hacer click con el botón derecho en el paso de retardo y si se selecciona la opción “Set Delay Time” se puede ajustar el número de milisegundos para la pausa (por defecto es 1000ms).

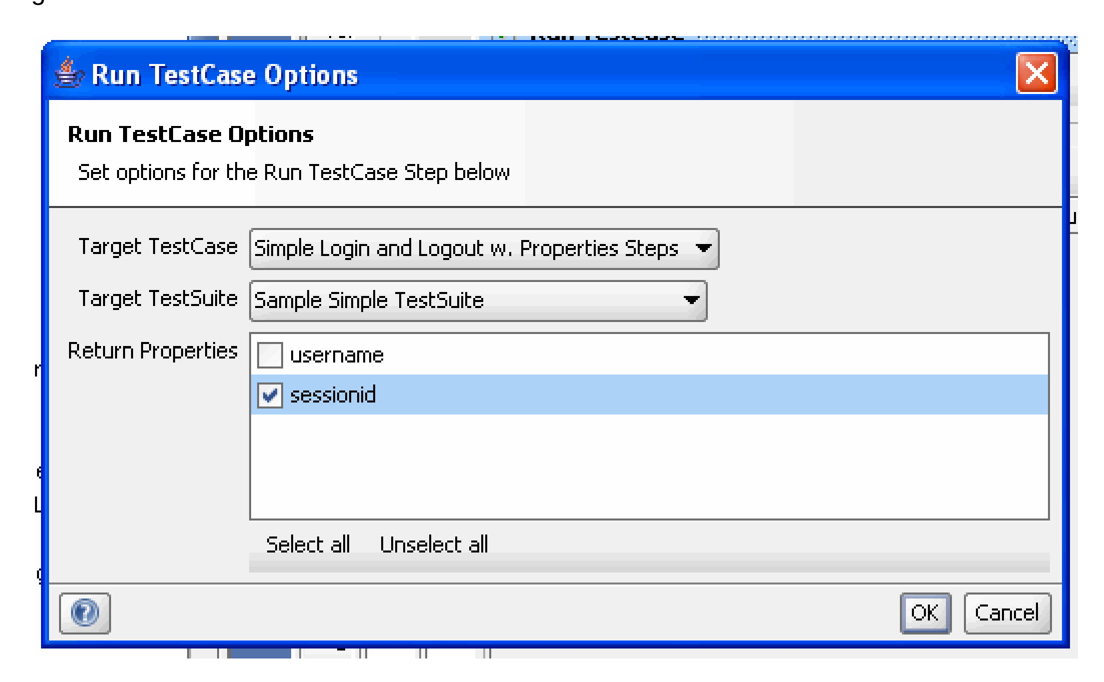


4.3.6. Run TestCase

El paso de ejecución de un TestCase permite ejecutar otro TestCase dentro del actual, opcionalmente se pueden ajustar las propiedades devueltas del TestCase destino.

Esto puede ser útil para dividir Testcase complejos en partes más pequeñas o compartir algunas pruebas funcionales entre Testcase.

Después de seleccionar "Run TestCase" cuando se crea un nuevo TestStep, se muestra el siguiente cuadro de diálogo:

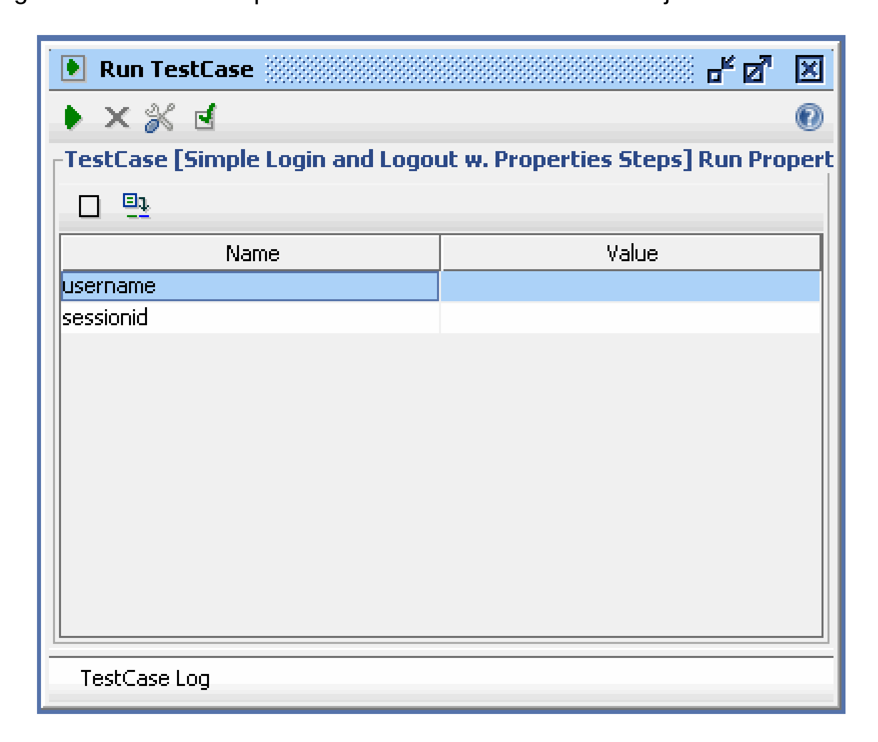


Seleccione el TestSuite y el TestCase de las listas desplegables. La lista de "Return Properties" contiene las propiedades del TestCase definidas para el TestCase destino. Seleccione las propiedades cuyos valores deben ser copiados y devueltos al paso de ejecución del TestCase después de ejecutarse.

El editor del paso de ejecución de un TestCase

Una vez creado, se abre el editor del paso de ejecución del TestCase que contiene (de arriba abajo)

- Una barra de herramientas con botones para ejecutar, cancelar, configurar y abrir el TestCase objetivo
- Una tabla que contiene las mismas propiedades que TestCase objetivo. Sólo los valores de las propiedades son editables.
- Un registro de TestCase que muestra la salida de la última ejecución



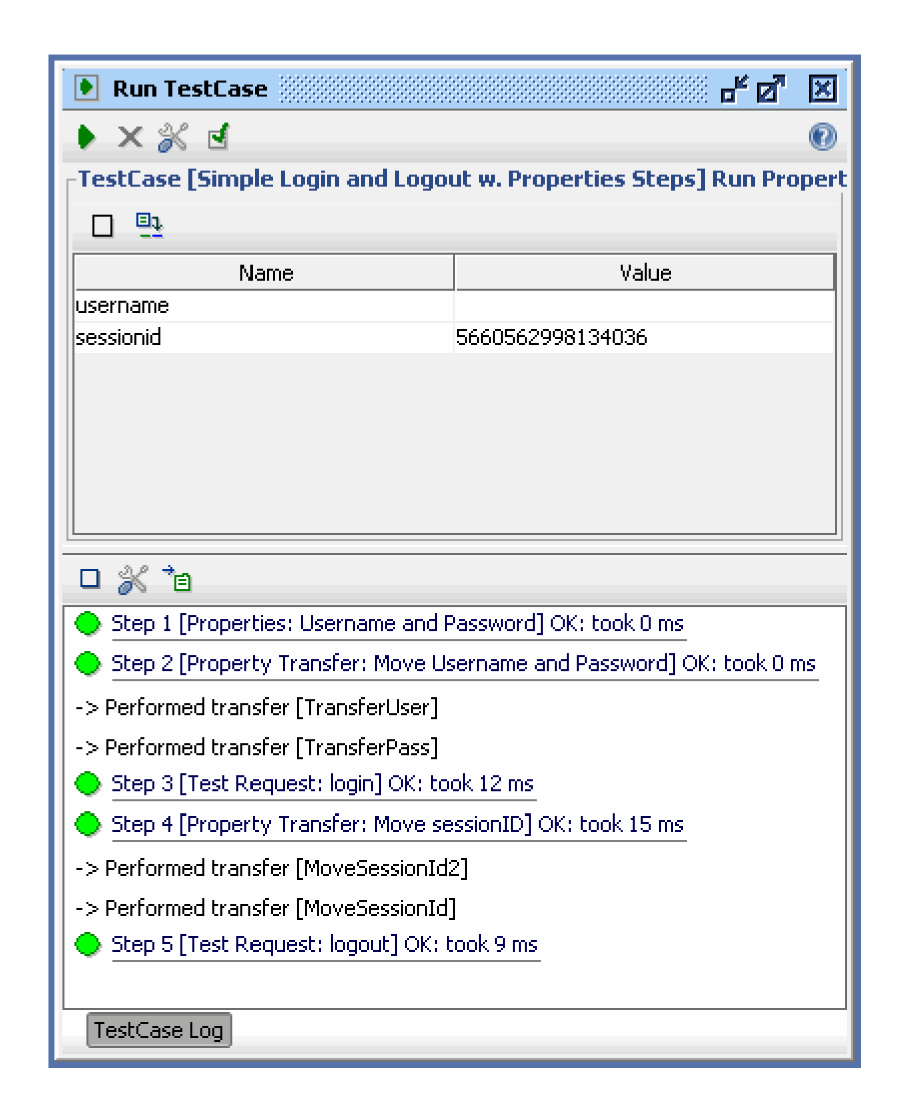
Ejecución

Cuando se ejecuta un TestCase, ocurre lo siguiente:

1. Los valores que contienen las propiedades y no están marcados como propiedades a devolver se copian a la TestCase objetivo
2. Se ejecuta el TestCase objetivo
3. Las propiedades marcadas como propiedades a devolver se copian del TestCase objetivo al paso de ejecución del TestCase
4. Si el TestCase objetivo falla, entonces también falla el paso de ejecución del TestCase

La propiedad manipulada en el paso 1 y 3 se puede observar como un argumento de entrada / salida para el TestCase, lo que permite su ejecución con diferentes valores. Esto podría ser combinado con un DataSource para la ejecución de un completo TestCase con un rango de valores de entrada.

Después de ejecutarlo, el registro del TestCase deberá contener la misma salida que se observa en el registro del TestCase para el TestCase objetivo.



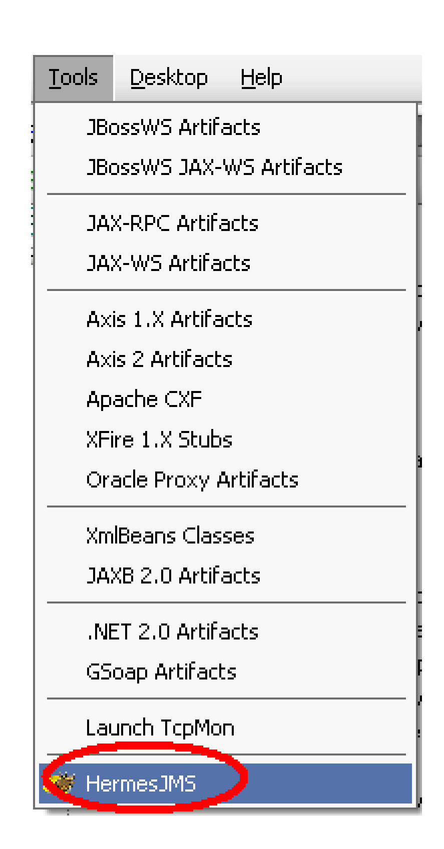
4.3.7. Pruebas JMS (HermesJMS)

La versión 3.5 de SoapUI, incluye soporte a las pruebas JMS, para cubrir esta funcionalidad se apoya en el proyecto HermesJMS.

HermesJMS es una consola extensible, que nos permite interactuar con proveedores JMS, hacienda fácil la navegación y la búsqueda de colas y "topics". Está plenamente integrado con JNDI permitiendo administrar objetos almacenados, crear sesiones JMS de las "connection factories" y usar cualquier destino definido.

Para obtener información adicional sobre el producto acceder a su página web:
http://www.hermesjms.com/confluence/display/HJMS/Home

Ejecución



En el menú de tools seleccionamos la opción HermesJMS que nos abrirá la utilidad externa.

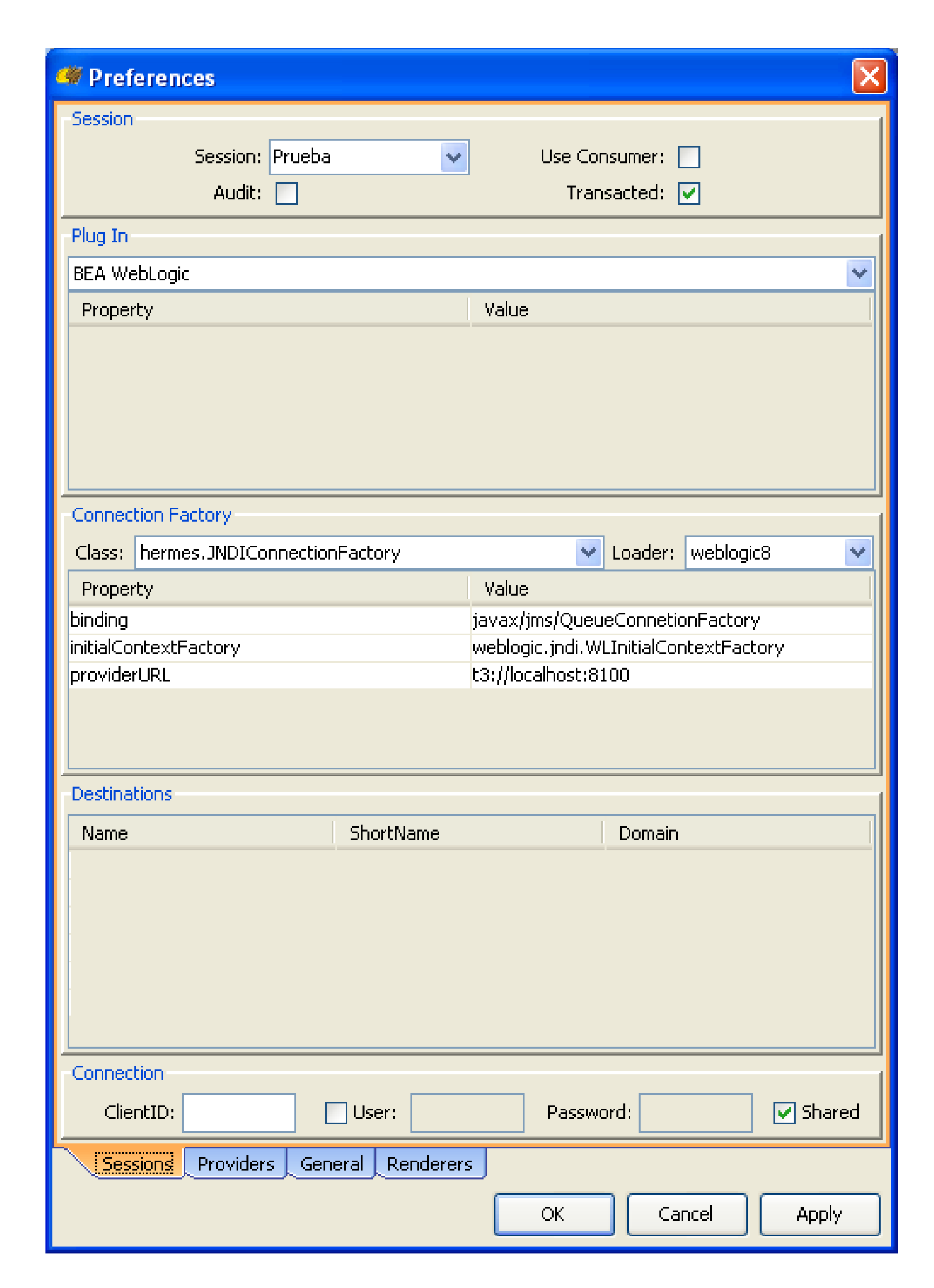
HermesJMS - \.hermes\hermes-config.xml
File Messages Actions Options Help
Sessions
jms
sessions
<new>
Files
Prueba
jms/internal/queue/WSDupsEliminationHistoryQueue
jms/internal/queue/WSStoreForwardQueue
q70/q70contextoGlobalForeignJMSTopic
q70/q70contextoGlobalJMSTopic
q70/q70desconexionUsuarioForeignJMSTopic
q70/q70desconexionUsuarioJMSTopic
contexts
stores
files
Prueba q70/q7...lobalJMSTopic

#	JMSM...	JMSD...	JMSTi...	JMST...	JMSR...	JMSC...	JMSE...	JMSPr...

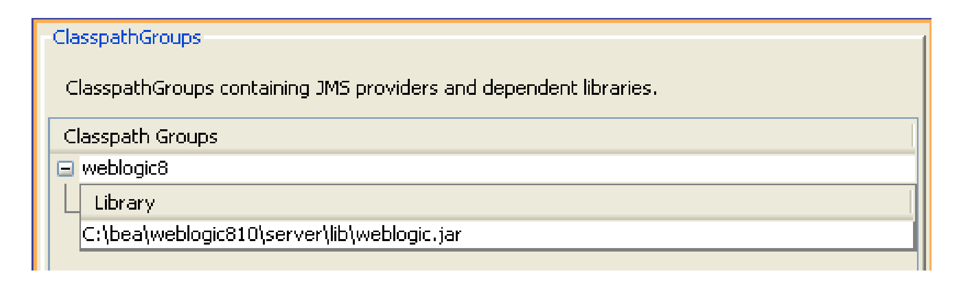
Filter:
Finished. No messages read.
Tools
Ready. 9:59:04 123M of 266M

Configuración weblogic8

Creamos un sesión nueva con la siguiente configuración



En la pestaña de providers, se deberá configurar las librerías necesarias para el servidor, en nuestro caso simplemente será necesario añadir weblogic.jar.

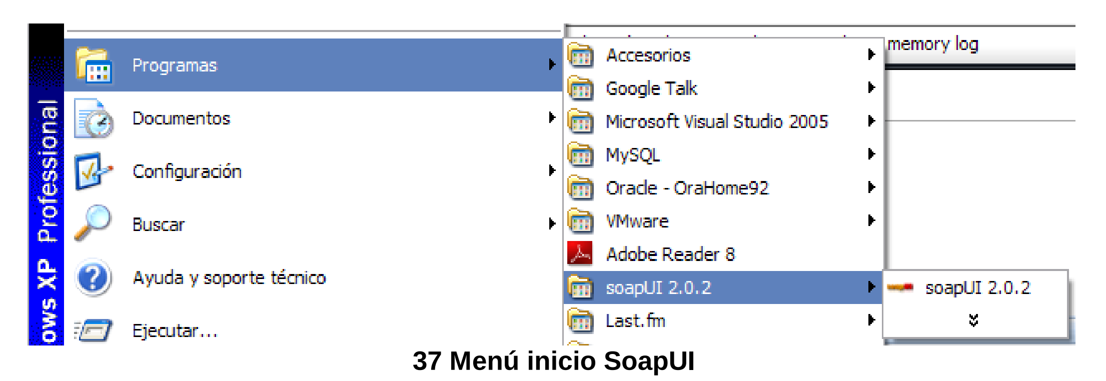


Para weblogic 11 pude ser necesario incluir en el path del entorno de ejecución el jar completo del Weblogic11 (hay una utilidad para conseguirlo)

5 Ejemplo de uso

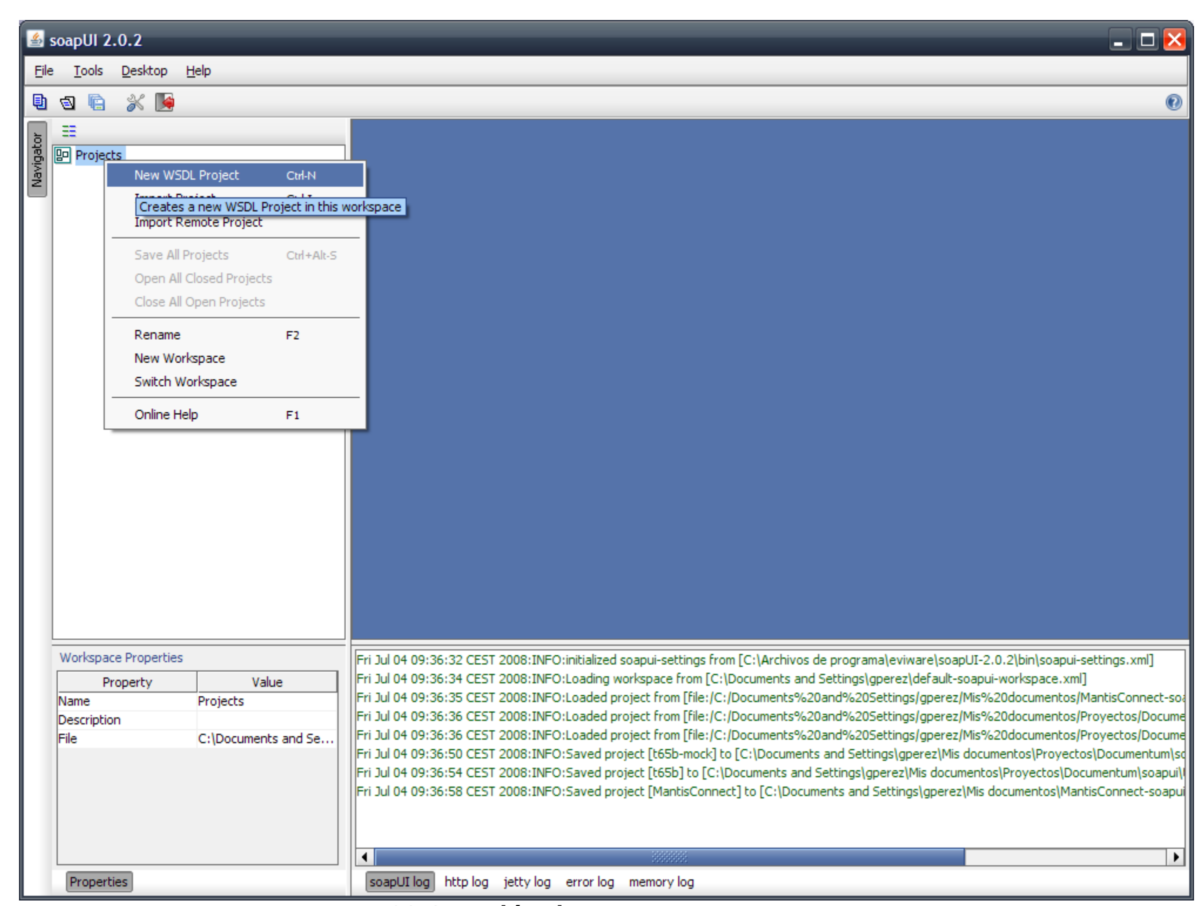
A continuación se describen los pasos necesarios para la crear un nuevo proyecto de servicio web en SoapUI.

1. Iniciar SoapUI



37 Menú inicio SoapUI

2. Crear un proyecto nuevo. Para ello tenemos dos opciones, click derecho sobre el nodo *"Projects"* del navegador y en el menú contextual hacer click sobre la opción *"new WSDL Project"* o bien Menú *"File"->"new WSDL Project"*

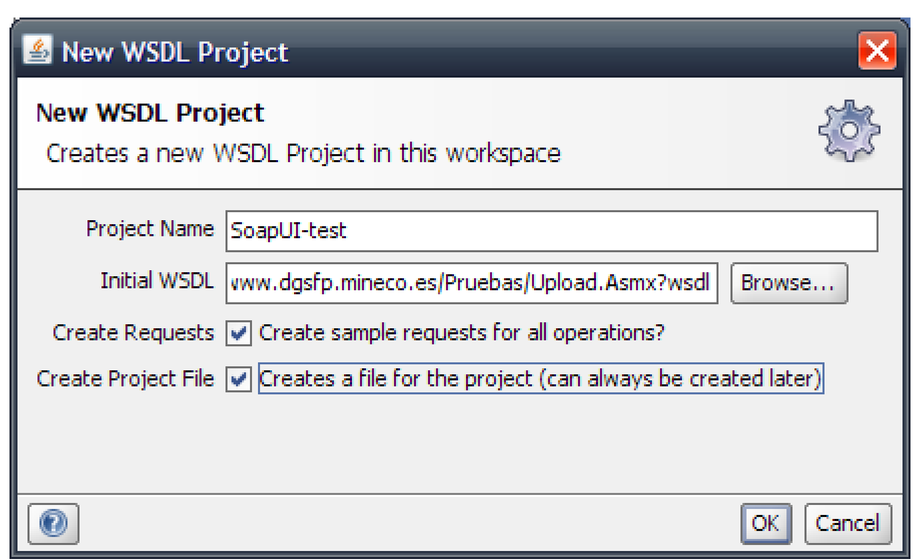


38 Creación de un proyecto nuevo

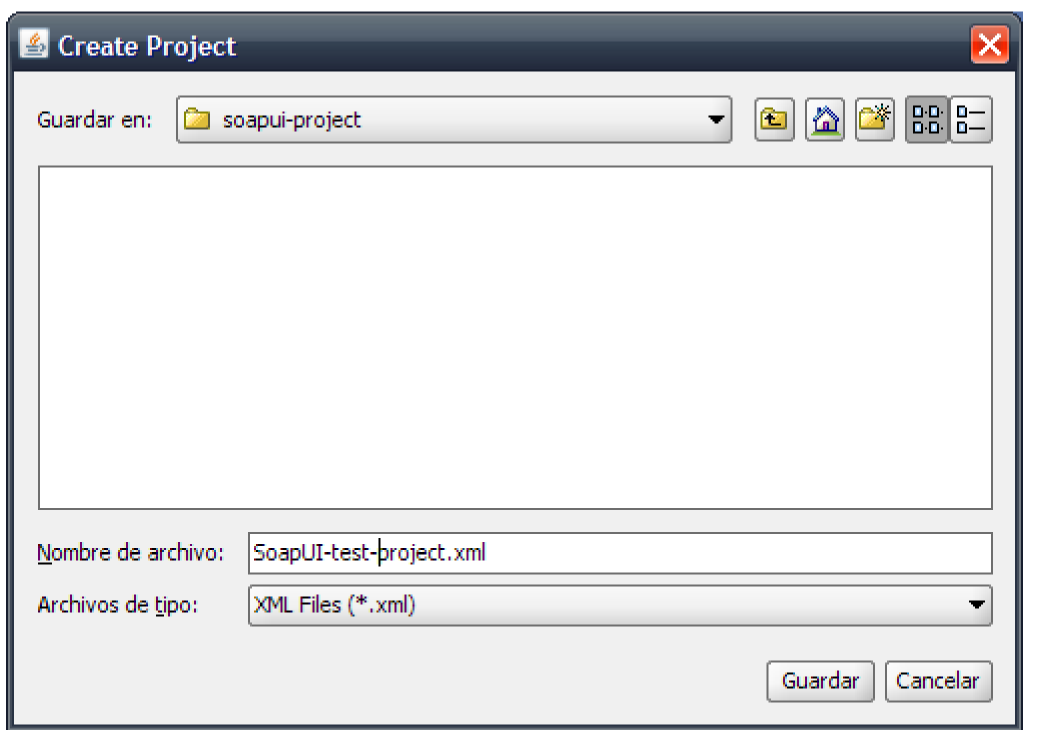
NOTA: Es posible que tanto la URL de invocación a los métodos del servicio web, como la URL de acceso al WSDL requieran autenticación por certificado. Si nos encontramos en ese caso, debemos adaptar la configuración SSL del SoapUI para que el asistente de nuevo proyecto termine satisfactoriamente.

3. Seguimos el asistente de creación de nuevo proyecto proporcionando los datos que se nos indiquen.

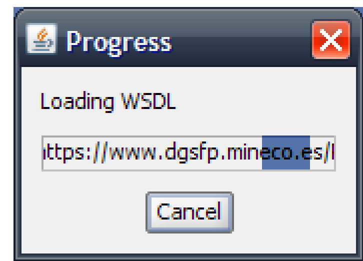
- El "*Project Name*" ponemos el nombre que queramos asignarle dentro del propio área de trabajo (*workspace*) del SoapUI.
- En "*Initial WSDL*" indicamos la URL hasta el fichero descriptor del servicio web. Es posible tanto indicar un fichero del sistema de fichero local como indicar una URL remota. Por ejemplo: https://www.dgsfp.mineco.es/Pruebas/Upload.Asmx?wsdl
- El checkbox "*Create Requests*" si se encuentra activado, le indica a SoapUI que debe autogenerar una petición de muestra para cada método publicado por el servicio web.
- El checkbox "*Create Project File*" si se encuentra activado hará que se nos abra una pantalla más en el asistente para indicar el nombre del fichero donde guardaremos la definición del proyecto. Este paso puede realizarse a posteriori desde los menús del soapUI.



39 Ventana asistente creación nuevo proyecto



40 Ventana de selección del nombre y ubicación del fichero de definición del proyecto



41 Ventana carga WSDL

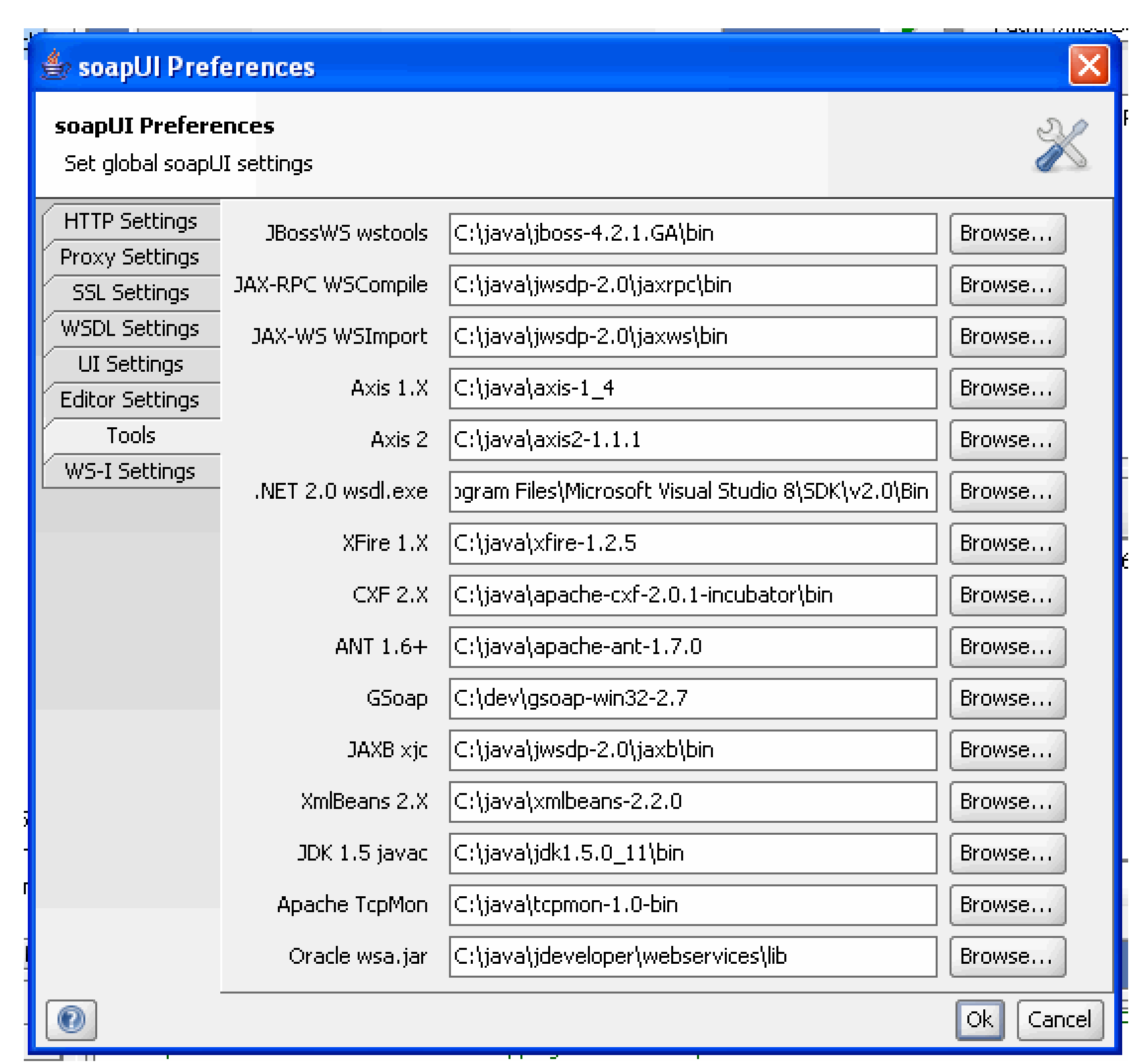
4. Una vez creado el proyecto. Podemos explorar los diferentes métodos que expone, así como lanzar las peticiones de prueba, una vez completemos los parámetros del que en cada caso requiera el servicio web.

6 Integración de herramientas

SoapUI puede ser integrado con diversas herramientas y *frameworks* de servicios web, incluyendo:

1. Herramientas de **generación de código** que permiten generación de clientes de un servicio web o de artefactos de implementación, a partir de un WSDL existente en un proyecto de SoapUI.
2. "WS-I Tools" integra la **validación** de servicios web contra "**WS-I Basic Profile**".
3. Apache TcpMon proporciona funcionalidad de **monitorización de tráfico HTTP**.

SoapUI no lleva incluido en la instalación las diferentes herramientas. Es necesario instalar individualmente cada una de ellas y posteriormente indicarle a SoapUI el *path* a la herramienta. Los diferentes paths se indican desde la pantalla de *Preferencias->Tools.* A continuación se muestra un ejemplo de como podría quedar dicha ventana:



42 Ejemplo de configuración de paths de herramientas externas

7 Casos de prueba prácticos

Se ha creado un proyecto de SoapUI 5.0 que recoge una serie de ejemplos que pueden resultar útiles para crear pruebas de integración de servicios Web.

El proyecto “Manual SoapUI” incluye para cada ejemplo descrito a continuación un caso de prueba. Cuando se carga en el programa, se inician un conjunto de servicios “Mock” que permiten lanzar el caso de prueba sin necesidad de tener un servicio arrancado, fuera del propio soapUI.

Los servicios “mock” son iniciados con la carga del script:

Manual SoapUI

Overview | TestSuites | WS-Security Configurations | Security Scan Defaults

Project Summary

File Path	D:\DATOS\JRIOBELL\My Documents\Manual-soapui-project.xml

Interface Summary

CurrencyConvertorSoap	http://www.webservicex.net/CurrencyConvertor.asmx?WSDL
CurrencyConvertorSoap12	http://www.webservicex.net/CurrencyConvertor.asmx?WSDL

Test Summary

TestSuites	1
TestCases	5
TestSteps	13
Assertions	4
LoadTests	0

SOAP Mock Summary

WsdlMockServices	4
WsdlMockOperations	4
WsdlMockResponses	5

REST Mock Summary

RestMockServices	0
RestMockActions	0

Edit — Load Script is invoked with log, project variables

```
mockServicesCount = project.getMockServiceCount()
for (i in 0..(mockServicesCount-1)) {
     project.getMockServiceAt(i).start();
     i++;
}
```

Description | Properties | Load Script | Save Script

Manual-soapui-project.xml.zip

7.1 Extracción de un elemento de un CDATA

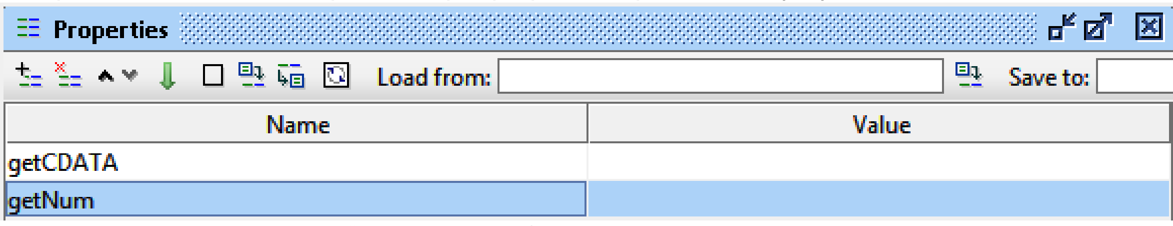
Objetivo: Extraer de una llamada un parámetro que está dentro de un CDATA para poder luego, por ejemplo, llamar a otro WebService.

Imaginemos que un servicio web devuelve un CDATA que contiene un xml que contiene un número. Necesitamos extraer ese número para llamar a otro servicio web y hacer una prueba.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://www.webserviceX.NET/">
   <soapenv:Header/>
   <soapenv:Body>
      <web:ConversionRateResponse>
         <web:ConversionRateResult><![CDATA[<persona><nombre>jorge</nombre><num>123</num></persona>]]>
         </web:ConversionRateResult>
      </web:ConversionRateResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

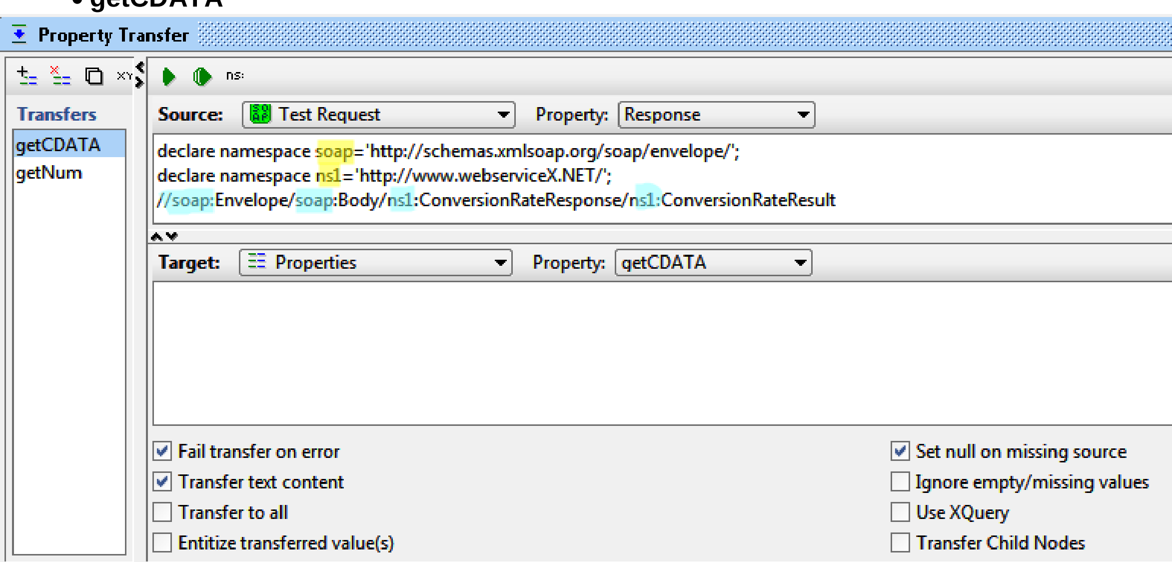
Es decir, queremos extraer **123** de esa respuesta.

Una posibilidad es crear un paso de tipo “properties” que defina 2 propiedades vacías:

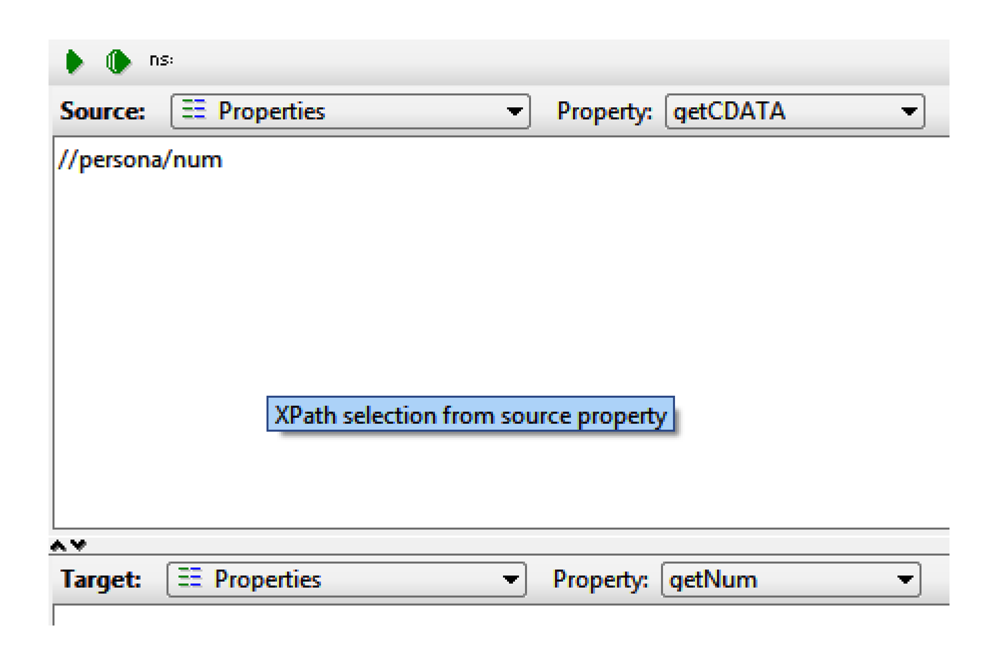


El siguiente paso es crear un “Propertie Trasfer” con 2 trasformaciones:

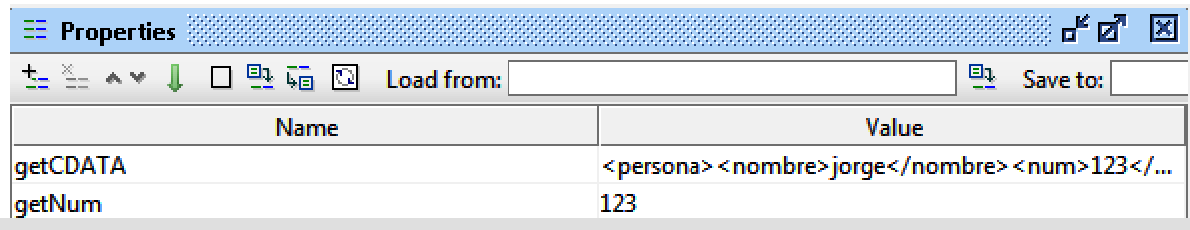
- **getCDATA**



- **getNum**



Ahora tendremos en la propiedad getNum el número que buscamos, que podremos usar en la hipotética petición posterior usando ${Properties#getNum}:



Properties

Load from: Save to:

Name	Value
getCDATA	<persona><nombre>jorge</nombre><num>123</...
getNum	123

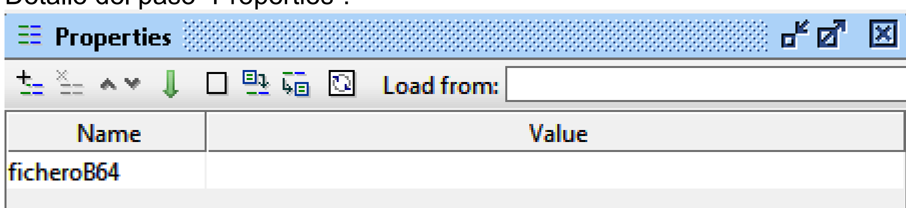
7.2 Codificar en Base64 un fichero y mandarlo como parámetro

Objetivo: Probar un Web Service que requiere un fichero codificado en Base64.

Imaginemos que tenemos que pasar un fichero codificado en Base64 para probar un Web Service. El fichero lo tenemos en nuestro disco duro.

La idea es crear un paso de tipo "Properties" que contenga una variable que seá el fichero en Base64. Para rellenar esa variable usaremos un paso de "Groovy".

Detalle del paso "Properties":



Properties

Load from:

Name	Value
ficheroB64	

Detalle del paso "Groovy":

```
import org.apache.commons.codec.binary.Base64

def groovyUtils = new com.eviware.soapui.support.GroovyUtils( contex


def file = new File('c:/windows/winhlp32.exe')
def fin = new FileInputStream(file)
def fileContent = new byte[(int)file.length()];
fin.read(fileContent)
log.info "fileContent= " + new String(fileContent)


def b64 = new Base64()
def encodeTextBytes = b64.encodeBase64(fileContent, true)

def fBase64 = new String(encodeTextBytes)
log.info "Fichero en Base64= "+ fBase64
groovyUtils.setPropertyValue("Properties", "ficheroB64", fBase64 )
testRunner.testCase.setPropertyValue( "ficheroB64" , fBase64)
```

Tendremos en la propiedad “ficheroB64” el contenido del fichero en Base64. Podemos usarlo en una hipotética petición usando ${ficheroB64}.

7.3 Decodificar un fichero codificado en Base64 que nos devuelve un servicio

Objetivo: Probar un Web Service que devuelve un fichero codificado en Base64. Luego, podemos usar el fichero devuelto para llamar a otro servicio…

Imaginemos que un servicio nos devuelve un string que contienen un fichero en base64, y queremos guardarlo en disco.

La idea es crear un paso “Groovy” para extraer y decodificar el fichero, que además, guarde en una propiedad el fichero codificado.

Decode Fichero en Base64

Script is invoked with log, context and testRunner variables

```
import org.apache.commons.codec.binary.Base64

def groovyUtils = new com.eviware.soapui.support.GroovyUtils( context )
def holder = groovyUtils.getXmlHolder( "Test Request#Response" )

holder.namespaces["soap"] = "http://schemas.xmlsoap.org/soap/envelope/"
holder.namespaces["ns1"] ="http://www.webserviceX.NET/"


def textBase64=holder.getNodeValues( "//soap:Envelope/soap:Body/ns1:ConversionRateResponse/ns1:ConversionRateResult" ).toString()

log.info "respuesta:" + textBase64
def b64 = new Base64()
def decodeTextBytes = b64.decode(textBase64.getBytes())
log.info "respuesta decodificada:" + new String(decodeTextBytes)

def responseFile = new java.io.File("D:/DATOS/cac.exe")
FileOutputStream fos = new java.io.FileOutputStream(responseFile)
fos.write( decodeTextBytes )
fos.flush()
fos.close()


groovyUtils.setPropertyValue("Properties", "ficheroB64", textBase64 )
testRunner.testCase.setPropertyValue( "ficheroB64" , textBase64)
```

7.4 Variables dinámicas

Objetivo: Realizar operaciones/modificaciones sobre posibles parámetros de un Web Service.

En este ejemplo se extrae un parámetro que es devuelto por un Web Service y después se define un nuevo parámetro que usa el anterior como máscara.

Varibales dinámicas

Script is invoked with log, context and testRunner variables

```
def groovyUtils = new com.eviware.soapui.support.GroovyUtils( context )
def holder = groovyUtils.getXmlHolder( "Test Request#Response" )

holder.namespaces["soap"] = "http://schemas.xmlsoap.org/soap/envelope/"
holder.namespaces["ns1"] ="http://www.webserviceX.NET/"


def vari=holder.getNodeValues( "//soap:Envelope/soap:Body/ns1:ConversionRateResponse/ns1:ConversionRateResult" ).toString()
if (vari.isNumber()){
     log.info vari.isNumber()
}

log.info "Variable extraida:" + vari

def temp = vari.replaceAll('\\[',' ')
temp = temp.replaceAll('\\]',' ')
log.info "temp:" + temp

temp=temp.toInteger()+1
log.info "temp + 1:" + temp

def date = new Date()
log.info "Objeto date vale: " + date
def int anio = date.getYear() as int
anio=anio+1900
log.info "AÑO:" + anio

def int mes = date.getMonth() as int
mes=mes +1
log.info "MES:" + mes
log.info "DIA:" + date.getDay()

def variModificada = temp.toString() + "AAAA" + mes.toString()
log.info variModificada
groovyUtils.setPropertyValue("Properties", "LaVariable", variModificada )
testRunner.testCase.setPropertyValue( "LaVariable" , variModificada)
```

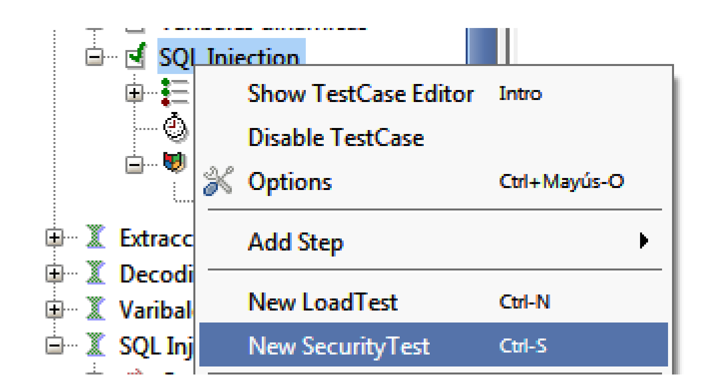
7.5 Prueba de seguridad [Inyección SQL]

Objetivo: Dar a conocer la utilidad de pruebas de seguridad que incorpora.

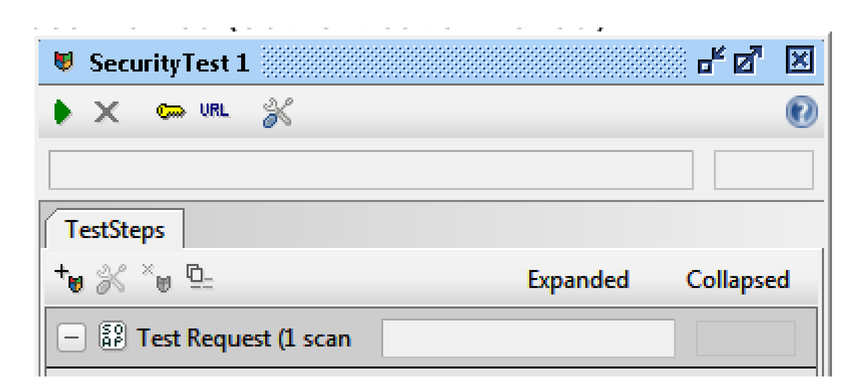
Imaginemos que queremos probar si un Web Service no presenta una vulnerabilidad de seguridad. Si atendemos a las vulnerabilidades más conocidas, en el caso de Web Services los principales ataques son intentados con peticiones específicas. Por ejemplo, la inyección SQL consiste en modificar la petición de manera que "colemos" sql en la ejecución del Web Service.

SopUI, en esta versión no comercial, nos facilita esta prueba.

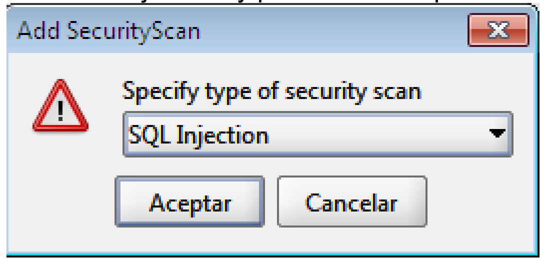
Para ello, sobre un caso de prueba funcional creado, podemos añadir una prueba de seguridad:



Al hacer doble click sobre el "Security Test 1" recién creado, veremos la petición del caso de prueba funcional, sobre la que podemos añadir pasos de seguridad, pulsando sobre el caso de prueba y pulsando el botón "   ":



Para este caso, seleccionamos "SQL Injection" y pulsamos "Aceptar":



Si pulsamos doble click sobre el paso, podremos editarlo:

SQL Injection

SQL Injection

Parameters:

Label	Name	XPath	Enabled
from	Request	declare namespace soap='http://schemas.xmlsoap.o...	☑
to	Request	declare namespace soap='http://schemas.xmlsoap.o...	☑

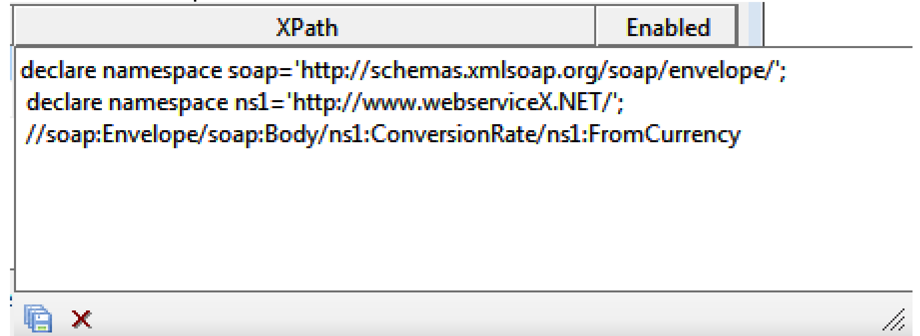
Strings for SQL injection can be changed under advanced settings

Assertions | Strategy | Advanced

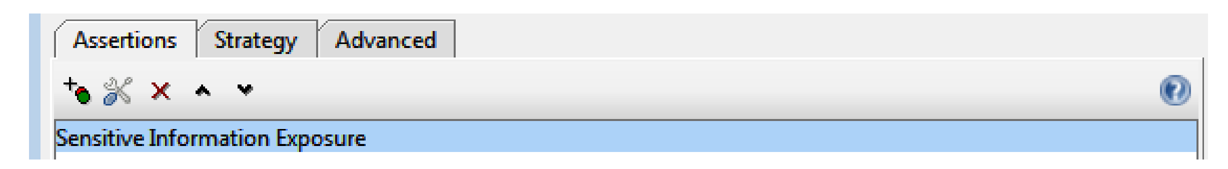
Sensitive Information Exposure

OK | Cancel

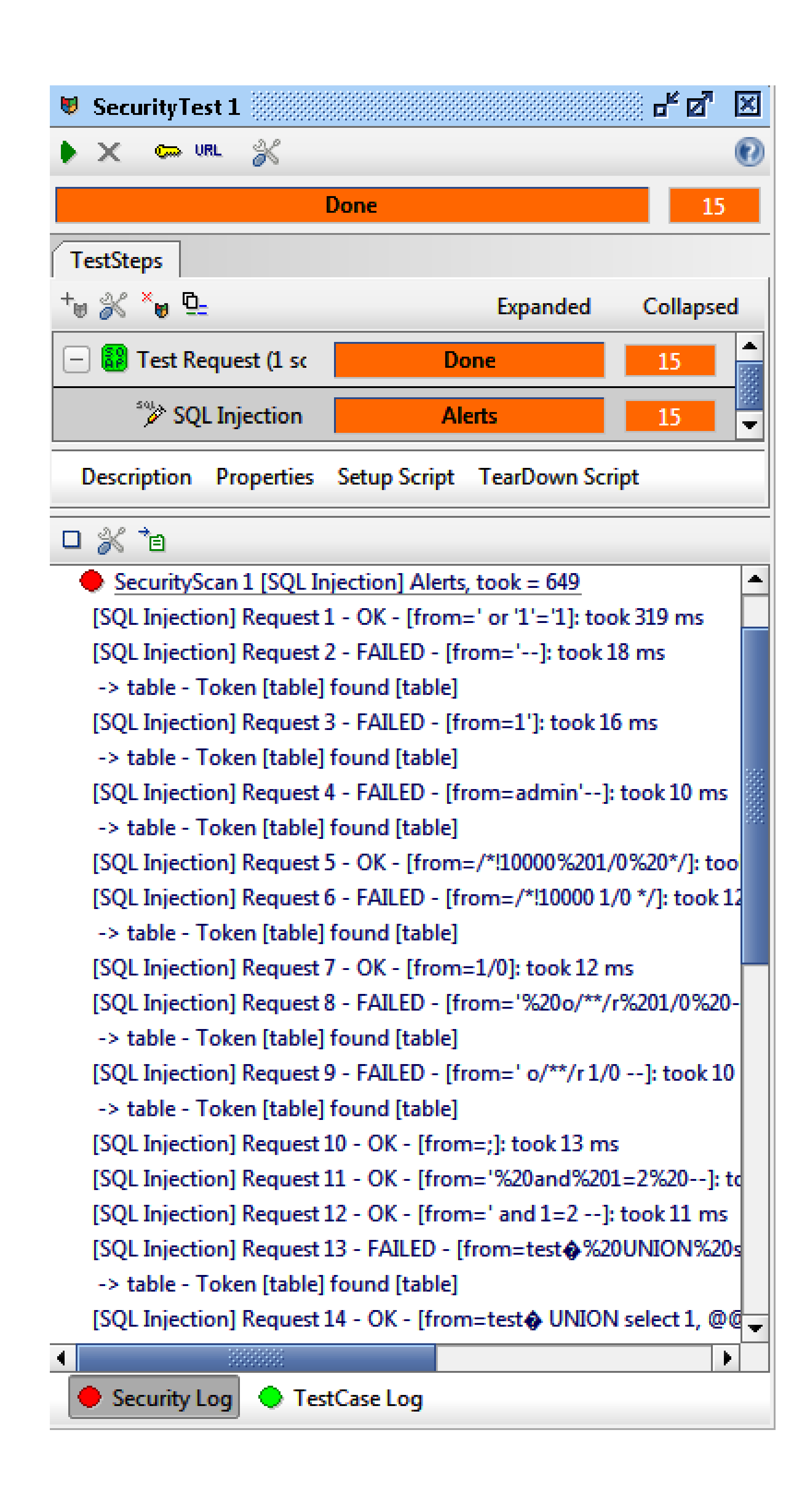
Lo que hace la herrmienta es presentar en la parte de arriba los parámetros a "modificar" para buscar la vulnerabilidad. En la versión de pago el propio soapUI crear las transfomraciones, pero en la no comercial tendremos que crearlas nosotros a mano:



En la pestaña "Advanced" vemos que tratará de incluir en esos parámetros para ver si se produce SL Injection. Para detectarlo, se crea una aserción, que en este caso, busca el token "table" en la respuesta.



Para el ejemplo, se ha creado un servicio “mock” que devuelve de manera aleatoria una respuesta con la cadena “table” , de manera que si ejecutamos el caso de prueba, veremos positivos:


SecurityTest 1
URL
Done
15
TestSteps
Expanded
Collapsed
Test Request (1 sc
Done
15
SQL Injection
Alerts
15
Description
Properties
Setup Script
TearDown Script
SecurityScan 1 [SQL Injection] Alerts, took = 649
[SQL Injection] Request 1 - OK - [from=' or '1'='1]: took 319 ms
[SQL Injection] Request 2 - FAILED - [from='--]: took 18 ms
-> table - Token [table] found [table]
[SQL Injection] Request 3 - FAILED - [from=1']: took 16 ms
-> table - Token [table] found [table]
[SQL Injection] Request 4 - FAILED - [from=admin'--]: took 10 ms
-> table - Token [table] found [table]
[SQL Injection] Request 5 - OK - [from=/*!10000%201/0%20*/]: too
[SQL Injection] Request 6 - FAILED - [from=/*!10000 1/0 */]: took 12
-> table - Token [table] found [table]
[SQL Injection] Request 7 - OK - [from=1/0]: took 12 ms
[SQL Injection] Request 8 - FAILED - [from='%20o/**/r%201/0%20-
-> table - Token [table] found [table]
[SQL Injection] Request 9 - FAILED - [from=' o/**/r 1/0 --]: took 10
-> table - Token [table] found [table]
[SQL Injection] Request 10 - OK - [from=;]: took 13 ms
[SQL Injection] Request 11 - OK - [from='%20and%201=2%20--]: tc
[SQL Injection] Request 12 - OK - [from=' and 1=2 --]: took 11 ms
[SQL Injection] Request 13 - FAILED - [from=test�%20UNION%20s
-> table - Token [table] found [table]
[SQL Injection] Request 14 - OK - [from=test� UNION select 1, @@
Security Log
TestCase Log